

**RANCANG BANGUN APLIKASI
GAME *BATTLESHIP MULTI PLAYER*
PADA JARINGAN LAN**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh:

M. FIKRI

10751000360



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2013**

RANCANG BANGUN APLIKASI GAME *BATTLESHIP* MULTI PLAYER PADA JARINGAN LAN

**M. FIKRI
10751000360**

Tanggal Sidang : 19 Juli 2013
Periode Wisuda : Oktober

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Battleship merupakan suatu game teka-teki yang membutuhkan logika sebagai dasar penyelesaiannya. Setiap teka-teki *Battleship* merepresentasikan sebuah laut dengan armada kapal-kapal tersembunyi yang memiliki orientasi kapal yang bisa diletakkan secara horizontal maupun vertikal di dalam kotak-kotak persegi. Game *battleship* ini pada umumnya hanya bisa dimainkan oleh satu komputer yang dimainkan oleh satu orang. Hal ini membuat para peminat *game battleship* Kurang menarik karena permainannya hanya bisa dimainkan satu orang. Berdasarkan hal itu dibangun sebuah game battleship multiplayer dengan menggunakan jaringan LAN sebagai media penghubung antara pemain game yang bertindak sebagai server dan client *Game battleship multiplayer* ini dirancang menggunakan OOA(*Object oriented analysis*) dan OOD(*object oriented design*). Pada game ini, setiap pemain berusaha untuk menghancurkan kapal – kapal perang pemain lawan dengan cara mengklik kotak – kotak yang dianggap sebagai letak dari kapal – kapal perang lawan. Setiap pemain hanya boleh menebak satu kali saja setiap gilirannya. Pemain yang masih menyisakan kapal perang yang memenangkan permainan game. Game ini diuji oleh 10 responden dengan metode *User Acceptance Test* (UAT). Hasil yang diperoleh dapat disimpulkan bahwa game ini dapat memberikan kesesuaian jawaban terhadap representasi pengetahuan yang diberikan dengan nilai persentase UAT sebesar 80% menyatakan bagus sehingga game ini dapat diterima oleh orang-orang yang sering main game.

Kata kunci : *Battleship*, *Game* ,*Lokal Area Network* (LAN).

KATA PENGANTAR



Assalamu 'alaikum wa rahmatullahi wa barakatuh.

Alhamdulillah, puji dan syukur senantiasa diucapkan ke hadirat Allah SWT, atas segala limpahan anugerah dan petunjuk-Nya, Tugas Akhir dengan judul **“RANCANG BANGUN APLIKASI GAME BATTLESHIP MULTI PLAYER PADA JARINGAN LAN”** ini dapat diselesaikan, sebagai salah satu syarat kelulusan dalam menyelesaikan studi di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.

Banyak sekali pihak yang telah membantu dalam penyelesaian Tugas Akhir ini, baik secara moril maupun materil. Untuk itu, terima kasih dihaturkan kepada:

1. Bapak Prof. DR. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Bapak Prof. Dra. Hj. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu DR. Okfalisa, ST, M.Sc., selaku Ketua Jurusan Teknik Informatika dan juga sekaligus Pembimbing Tugas Akhir terima kasih atas waktu, tenaga, pikiran, saran dan bimbingan yang telah diberikan.
4. Ibu Elin Hearani, ST, M.Kom selaku Sekretaris Jurusan Teknik Informatika. sekaligus selaku Penguji I Tugas Akhir yang telah memberikan masukan yang bermanfaat kepada penulis
5. Bapak Surya Agustian, ST, M.Kom selaku Penguji II Tugas Akhir yang telah memberikan masukan yang bermanfaat kepada penulis
6. Bapak Reski Mai Candra, ST, M.Sc selaku Koordinator Tugas Akhir.

7. Seluruh dosen dan karyawan Fakultas Sains dan Teknologi, khususnya Jurusan Teknik Informatika.
8. Kedua orang tuaku tercinta Sudirman dan Rosnani yang selalu memberi semangat, dukungan dan doa kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
9. Adik-Adiku tersayang yang telah memberikan motivasi dalam menyelesaikan tugas akhir ini.
10. Herlina, SE. Sy yang telah memberikan motivasi dan bantuannya dalam menyelesaikan tugas akhir ini.
11. Teman-teman Jurusan Teknik Informatika khususnya angkatan 2007, terima kasih atas dukungan, saran, kritik dan diskusinya untuk kesempurnaan penyusunan Tugas Akhir ini.
12. Teman-teman satu kos, Yogi, S. Pt, Jhon, S. Pt, Firman, S.Sy, Dimas, S. Pt Razak, Damri terima kasih atas dukungan, saran, kritik dan diskusinya untuk kesempurnaan penyusunan Tugas Akhir ini.
13. Semua pihak yang tidak bisa disebutkan satu persatu yang telah banyak membantu selama ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, kritik serta saran yang membangun dari rekan-rekan pembaca sangat dibutuhkan agar dapat membuat Tugas Akhir ini lebih baik. Akhir kata penulis berharap agar Tugas Akhir ini bisa memberikan manfaat bagi pembaca dan semua pihak yang berkepentingan. Terima kasih.

Pekanbaru, 19 Juli 2013

Penulis

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-2
1.3 Batasan Masalah.....	I-2
1.4 Tujuan	I-2
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI	II-1
2.1 <i>Game</i>	II-1
2.2.1 Pengertian <i>Game</i>	II-1
2.2.2 Jenis <i>Game</i>	II-3
2.2.3 Sejarah <i>Game</i>	II-4
2.2.4 Pembuatan <i>Game</i>	II-6
2.2.5 Klasifikasi <i>Game</i>	II-7

2.2	<i>Battleship</i>	II-9
2.3	<i>Visual Basic</i>	II-12
2.3.1	<i>Game ular tangga</i>	II-13
2.4	Perangkat Lunak Berorientasi Objek	II-13
2.4.1	OOA (<i>Objek Oriented Analysis</i>)	II-13
2.4.2	OOD (<i>Objek Oriented Desagn</i>)	II-14
2.5	UML (<i>Unifield Modelling Language</i>).....	II-15
2.5.1	Konsep dasar UML	II-16
2.6	Jaringan Komputer	II-23
2.7	Arsitektur Jaringan Komputer.....	II-24
2.7.1	Jaringan <i>Peer to Peer</i>	II-24
2.7.2	IP Address	II-25
BAB III METODOLOGI PENELITIAN.....		III-1
3.1	<i>Flowchart</i> Metodologi Penelitian	III-1
3.2	PerumusanMasalah	III-2
3.3	PengumpulanData	III-2
3.3.1	Jenis data	II-2
3.3.2	Teknik pengumpulan data	II-2
3.4	Analisa dan Perancangan	III-3
3.5	Implementasi dan Pengujian	III-4
3.6	Kesimpulan dan Saran.....	III-5
BAB IV ANALISIS DAN PERANCANGAN		IV-1
4.1	Analisa Sistem Lama.....	IV-1
4.2	Analisa Sistem Baru	IV-3
4.2.1	Arsitektur Aplikasi	IV-6
4.2.2	Perancangan Animasi dan Suara.....	IV-7
4.2.3	Pengaturan Koneksi Jaringan	IV-8
4.2.4	Perancangan <i>Interface</i> untuk Menyusun Kapal	IV-12
4.2.4.1	Perancangan Medan Perang.....	IV-12

4.2.4.2	Penentuan Posisi Kapal.....	IV-13
4.2.4.3	Pengecekan Posisi Tembakan Kapal	IV-13
4.3	Cara bermain <i>Game battleship multyplayer</i>	IV-13
4.3.1	Penentuan pemenang	IV-13
4.3.2	Perhitungan <i>Score</i>	IV-14
4.4	Perancangan Sistem	IV-14
4.4.1	<i>Use Case Diagram</i>	IV-15
4.4.2	<i>Activity Diagram</i>	IV-15
4.4.3	<i>Sequence Diagram</i>	IV-17
4.5	Perancangan Struktur Menu	IV-19
4.5.1	Tampilan <i>Splash Screen</i>	IV-19
4.5.2	Tampilan Pengaturan Koneksi	IV-20
4.5.2.1	Tampilan perancangan menu <i>Server</i>	IV-20
4.5.2.2	Tampilan perancangan menu <i>Client</i>	IV-21
4.5.3	Tampilan Tempat Penyusunan Kapal	IV-22
4.5.4	Tampilan Medan Perang	IV-23
BAB V IMPLEMENTASI DAN PENGUJIAN		V-1
5.1	Implementasi Sistem	V-1
5.5.1	Lingkungan Implementasi.....	V-1
5.5.2	Kebutuhan Perangkat Keras	V-1
5.5.3	Kebutuhan Perangkat Lunak	V-2
5.2	Tampilan Menu Utama	V-2
5.3	Pengujian Sistem.....	V-5
5.3.1	Pengujian <i>Back Box</i>	V-5
5.3.2	Pengujian Menu koneksi	V-6
5.3.3	pengujian infut form nama	V-7
5.3.4	pengujian tampilan medan perang	V-8
5.3.5	Kesimpulan pengujian dengan <i>black box</i>	V-8
5.4	pengujian system kepada User (<i>User Acceptance Test</i>)	V-9

5.4.1 Kesimpulan pengujian User (<i>User Acceptance Test</i>)	V-7
BAB VI PENUTUP	VI-1
6.1 Kesimpulan	VI-1
6.2 Saran.....	VI-1
DAFTAR PUSTAKA	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dekade terakhir, tingkat kemajuan teknologi komputer sangat pesat. Permainan (*game*) bermunculan seiring dengan perkembangan komputer tersebut. *Game* menjadi salah satu opsi untuk menghilangkan kejenuhan akibat aktivitas yang dilakukan oleh seseorang. Pada saat ini, *game* tidak hanya dapat dimainkan oleh satu komputer saja. Namun, *game multi-user* juga dapat dibuat dan dimainkan. Faktanya adalah *game multi-user* lebih diminati oleh para *gamers*.

Dalam memainkan suatu *games* oleh beberapa komputer, maka dibutuhkan suatu media penghubung antara komputer yang satu dengan komputer lainnya. Media penghubung yang sering digunakan adalah *Local Area Network* (LAN). LAN adalah jaringan lokal antar komputer dalam suatu tempat yang cakupannya tidak terlalu luas, seperti antar ruangan.

Permasalahan *Game battleship* ini pada umumnya hanya bisa dimainkan oleh satu komputer yang dimainkan oleh satu orang. Hal ini membuat para peminat game merasa Kurang menarik dengan permainannya dengan tampilan game yang sangat sederhana. Berdasarkan hal itu dibangun sebuah *game battleship multiplayer* dengan menggunakan jaringan LAN sebagai media penghubung antara pemain game yang bertindak sebagai server dan client *Game battleship multiplayer* ini dirancang menggunakan OOA (*Object oriented analysis*) dan OOD(*object oriented design*). Permainan ini menyediakan beberapa buah tempat lokasi yang berbentuk blok-blok sebagai medan perangnya. Kapal-kapal tersebut mempunyai ukuran yang berbeda serta disusun dalam medan perang masing-masing. Letak kapal-kapal tersebut hanya dapat dilihat pada komputer masing-masing dan tidak dapat dilihat pada komputer lawan. Setiap pemain dapat meletakkan kapal-kapal tersebut kedalam blok-blok yang telah disediakan secara vertikal maupun horizontal.

Setiap pemain berusaha untuk menghancurkan kapal lawan dengan cara meng-klik kotak yang dianggap dari kapal lawan. Setiap pemain hanya diperbolehkan meng-klik satu kali.

Berdasarkan uraian diatas, maka pada tugas akhir ini dilakukan penelitian tentang rancang bangun aplikasi game *battleship multi player* Pada jaringan LAN. Abraham Ranardo Sumarsono (2009) melakukan penelitian sejenis yaitu membuat aplikasi game *battleship*. Perbedaanannya penelitian tugas akhir ini dengan penelitian Abraham Ranardo Sumarsono adalah membuat game *battleship* yang menggunakan satu komputer, sedangkan pada penelitian tugas akhir ini *battleship* dapat dimainkan dengan menggunakan jaringan komputer dan dapat dimainkan oleh dua pemain dengan menggunakan dua komputer.

Berdasarkan beberapa faktor yang telah disebutkan diatas, maka pada penelitian ini diberi judul “Rancang Bangun Aplikasi Game *Battleship Multi Player* Pada Jaringan LAN.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka yang menjadi rumusan masalah pada penelitian ini adalah bagaimana membuat sebuah permainan strategi *battleship* yang dapat dimainkan dengan menggunakan jaringan LAN.

1.3 Batasan Masalah

Agar pembahasan tidak menyimpang dari pokok perumusan masalah, maka penulis membatasi penelitian yang dilakukan yaitu:

1. Jumlah pemain sebanyak 2 orang.
2. Rincian dari masing – masing kapal perang, yaitu :
 - a. *Battleship* dengan ukuran 4 kotak sebanyak 1 buah.
 - b. *Cruiser* dengan ukuran 3 kotak sebanyak 2 buah.
 - c. *Submarine* dengan ukuran 2 kotak sebanyak 3 buah.
 - d. *Destroyer* dengan ukuran 1 kotak sebanyak 4 buah.
3. Ukuran dari medan perang adalah 10 kotak x 10 kotak.

1.4 Tujuan

Tujuan dari penelitian ini adalah merancang dan membangun sebuah *game* strategi *battleship Multi Player* menggunakan jaringan LAN.

1.5 Sistematika Penulisan

Sistematika penulisan bertujuan memudahkan dalam pemahaman permasalahan secara detail tugas akhir. Sistematika penulisan penelitian tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang deskripsi umum dari tugas akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang teori-teori menyeluruh mengenai metode yang digunakan penyelesaian penelitian tugas akhir.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang cara yang dilakukan untuk melakukan penelitian dan mendapatkan data guna membuat *game* strategi *battleship*.

BAB IV ANALISA DAN PERANCANGAN

Bab ini menjelaskan tentang proses merancang aplikasi *game battleship*.

BAB VI IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan tentang cara yang dilakukan untuk melakukan penelitian dan mendapatkan data guna membuat *game* strategi *battleship*.

BAB VI PENUTUP

Bab ini menjelaskan tentang kesimpulan dan saran dari penelitian yang dilakukan.

BAB II

LANDASAN TEORI

2.1 *Game*

Game atau lebih tepatnya disebut video game adalah permainan yang menggunakan interaksi dengan antarmuka pengguna melalui gambar yang dihasilkan oleh piranti video. Permainan video umumnya menyediakan sistem penghargaan – misalnya skor – yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan

2.1.1 *Pengertian Game*

Kata *game* berasal dari bahasa inggris. Dalam kamus bahasa indonesia istilah “*game*” adalah permainan. Permainan merupakan bagian dari bermain dan bermain juga bagian dari permainan yang kesuanya saling berhubungan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*intellectual playability game*) yang juga bisa diartikan sebagai arena keputusan dan aksi permainannya.

Game komputer adalah sebuah program *software* diman satu atau lebih pemain berusaha untuk membuat keputusan lewat kontrol terhadap *object* dan *resource* guna memenuhi satu tujuan tertentu.

Video game adalah *game* yang berbasis elektronik dan visual. *Video game* diaminikan dengan memanfaatkan media visual elektronik yang biasanya menyebabkan radiasi pada mata, sehingga mata pun lelah dan biasanya diiringi dengan sakit kepala

Teori permainan (game) pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi:

“Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk

meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.

Menurut Agustinus Nilwan dalam bukunya “Pemrograman Animasi dan Game Profesional” terbitan Elex Media Komputindo, game merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaan animasi haruslah memahami pembuatan game. Atau jika ingin membuat game, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan. (Dawang Muchtar, 2005)

Menurut Jim (2007) game merupakan aktifitas terstruktur atau semi terstruktur yang biasanya bertujuan untuk hiburan dan kadang dapat digunakan sebagai sarana pendidikan. Karakteristik game yang menyenangkan, memotivasi, membuat kecanduan dan kolaboratif membuat aktifitas ini digemari oleh banyak orang. Game merupakan sebuah permainan yang menarik dan menyenangkan. Game merupakan fenomena global. Permainan elektronik yang menggunakan media *computer*, *phone* *seleluller* maupun konsol seperti playstation atau x-box sudah menjamur kemana-mana. Bisnis game juga sudah merambah kemana-mana, namun ironisnya konten dari game sebagian besar berisi hiburan dan sangat sedikit yang berkonten pendidikan (edukasi). Sebenarnya tanpa disadari game dapat mengajarkan banyak ketrampilan dan game dapat dijadikan sebagai salah satu alternatif pendidikan. Game yang ada sekarang memiliki berbagai macam jenis atau yang dikenal dengan istilah genre. Game bergenre edukasi ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain. Sehingga dengan perasaan senang diharapkan anak bisa lebih mudah memahami materi pelajaran yang disajikan. Genre ini sebenarnya lebih mengacu kepada isi dan tujuan game, bukan genre yang sesungguhnya.

Pengertian *game* menurut beberapa ahli adalah:

- a. Menurut Chris, (1982) : *game* memiliki empat sifat. Yaitu: sistem formal, tertutup (ini artinya *game* memiliki peraturan; "formal" berarti bisa didefinisikan, bukan mengenakan jas dan dasi); mengandung interaksi; melibatkan konflik; dan aman untuk dilakukan. Paling tidak dibandingkan hal yang diwakilinya (contohnya, *football* Amerika bukanlah hal yang benar-banar aman, cedera sering terjadi tapi sebagai *game* ini adalah perwakilan abstrak dari perang, dan jelas lebih aman dari pada menjadi tentara di medan tempur).
- b. Menurut Tracy. (2008) *Game* adalah sesuatu yang memiliki "akhir dan cara mencapainya" artinya ada tujuan, hasil dan serangkaian peraturan untuk mencapai keduanya.
- c. Menurut Barton. (2009) seorang sosiolog Perancis, dalam bukunya berjudul *Les jeux et les hommes* menyatakan *game* adalah aktivitas yang mencakup karakteristik berikut: fun (bebas bermain adalah pilihan bukan kewajiban).
- d. Menurut Dawang Muctar, 2005: *Game* atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan refreshing.

2.1.2 Jenis *Game*

Sebuah PC tidak akan sempurna tanpa *game*, ungkapan tersebut menggambarkan keadaan saat ini dimana hampir setiap PC sudah memiliki *game*, baik *game* berukuran besar maupun kecil.

Game yang dipasang di PC dapat diklasifikasikan menjadi beberapa tipe, yaitu:

- a. *Side Scrolling Game*

Pada jenis *game* ini karakter dapat bergerak kesamping dapat diikuti dengan gerakan background. Contoh *game* dari tipe ini adalah Super Mario, Metal Slug, dan sebagainya.

- b. *Shooting Game*

Ada beberapa *game shooting*, diantaranya adalah First Person Shooting. Pada dasarnya kedua game tersebut sama, yaitu pemain menembak musuh. Contoh *game* dari tipe ini adalah Counter Strike, Doom, Chicken Shoot, dan sebagainya.

c. *RPG (Role Play Game)*

Adalah *game* memainkan peran suatu karakter dalam menjalankan misi tertentu. Contoh dari *game* ini adalah Final Fantasy, Lord Of The Ring, Shining Force, dan sebagainya.

d. *RTS (Real Time Strategy)*

Pada *game* ini pemain memerintahkan beberapa karakter untuk melakukan aksi tertentu. Sebagian besar game RTS adalah game strategi perang seperti Red Alert 2, Comando, War Craft, dan sebagainya.

e. *Simulation*

Game ini merupakan simulasi dari keadaan riil. Ada beberapa pembagian game simulasi antara lain game simulasi manusia seperti The Sims.

f. *Racing*

Contoh dari *game racing* antara lain: Need For Speed, Grand Turismo, FI, dan sebagainya.

g. *Fighting*

Dalam *game fighting* pemain memainkan satu atau lebih karakter untuk bertarung dengan karakter lain, contoh dari game ini antara lain: Tekken, Street Fighter, King of Fighter, Battleship dan sebagainya.

2.1.3 Sejarah Game

Game dan keberadaan manusia tidak dapat dipisahkan, membicarakan mengenai game akan selalu berkaitan dengan adanya manusia, begitu juga sebaliknya, keduanya eksis dan saling mempengaruhi satu sama lain. Manusia kebudayaan yang bermain, pada dasarnya manusia memiliki hasrat untuk bermain. Hasrat natural tersebut kemudian dimunculkan dalam bentuk sebuah *game*,

bentukan manusia (*culture*). Manusia bermain karena adanya interaksi antar sesamanya, interaksi dan gerak yang merupakan kebutuhan dasar dari manusia. Naluri bermain manusia sudah ada semenjak bayi, permainan sederhana seperti *ci-luk-ba*, *petak umpet* saat manusia beranjak balita kesemuanya merupakan bagian dari naluriah manusia, hal tersebut dikarenakan *game* atau permainan, memberikan kegunaan (mengasah kemampuan seperti berfikir strategi) dan kesemuanya adalah untuk kesenangan. Kemunculan game kedunia memiliki sejarah yang cukup panjang, banyak unsur yang telah mempengaruhi eksistensinya, hingga game telah menjadi industri besar yang cukup menjanjikan pada masa ini. Ilmu pengetahuan yang telah memproduksi teknologi telah banyak mempengaruhi perkembangan game. Berikut pembahasan game di awal kemunculannya kedunia hingga sepak terjangnya di dunia industri baru saat ini. Chris, (1982).

1. Generasi pertama

Game yang terkenal pada masa itu adalah PONG merupakan game yang seperti bermain ping pong yang dapat dimainkan oleh dua orang. Pada waktu PONG dimainkan pada console Odyssey yang di produksi pertama kali oleh perusahaan Magnavox. Console ini sangat pencuri perhatian karena ini merupakan terobosan pertama sebuah permainan yang dapat dimainkan oleh manusia yang berinteraksi dengan mesin. Dan benar dapat mendapatkan kesenangan dari permainan yang dimainkan. Console ini keluar pada tahun 1972.

2. Generasi kedua

Generasi kedua dimulai pada tahun 1976 dimana era munculnya 8 bit kurang dari 4 bit era. Pada generasi ini keluar console game yang di rilis oleh Fairchild Channel F dan Radofin 1292 Advanced Programmable Video system. Dan pada generasi kedua ini juga banyak bermunculan console video game dan menjadi buruan orang banyak. Console yang termasuk generasi kedua adalah dari ki-ka adalah Fairchild Channel F, Atari 2600, Magnavox Odyssey ver. 2, Atari 5200.

3. Generasi ketiga

Merupakan generasi munculnya Nintendo dan Sega. Console game yang pertama kali saya mainkan. Generasi tiga ini dimulai pada tahun 1983 yang di produksi oleh negeri matahari terbit yaitu Jepang yang perusahaannya bernama Famicom

atau yang suka saya sebut sebagai nitendo yang juga menggunakan 8 bit prosessor. Dan setelah itu muncul SEGA yang menjadi saingan NITENDO pada game console yang beredar. Secara kualitas gambar kedua console game ini hampir sama, tetapi sega memiliki 16 bit prosessor .

4. Generasi Keempat

Disini mulai mengenal console game mini yang bisa dibawa kemana-mana. Generasi empat dimulaia pada tahun 1988 dimana nitendo dan sega masih terus bersaing Sega mengeluarkan sega baru yaitu Sega mega drive yang kualitas gambarnya lebih bagus dibandingkan nitendo setelah itu NES tidak mau kalah sehingga NES kembali meluncurkan SNES(super nitendo entertainment system. Dan setelah itu bermunculan console game mini seperti gameboy yang diprduksi oleh nitendo, dan mulai berkeluaran produk console game yang berbasis CD.

5. Generasi Kelima

Ini merupakan perkembangan yang paling drasitis SONY mengeluarkan SONY PlayStation yang merupakan console game yang sudah bisa memiliki gambar 3D dan menggunakan prosessor 32 bit. Console ini menggunakan CD untuk memainkan jenis permainan dengan menggunakan optic untuk membaca kepingan CD setiap game yang di masukan. Kualitas gambar hamper seperti real. Seluruh orang didunia menganggap ps ini sebagai trobosan yang baik dalam dunia game.

6. Generasi keenam

Pada Generasi ini mulai berdatangan console game yang mendukung gambar 3D dan mulai persaingan yang ketat karena beberapa produsen menunjukan keunggulan masing- masing dari console yang mereka produksi. Seperti xbox yang ingin menyaingi ketenaran dari playstation tetapi SONY tidak mau ambil diam, lalu dia memproduksi PS2 yang kualitas gambar lebih baik dari ps1. Ps2 ini menggunakan RAM dan menggunakan kepingan DVD karena data dari video game yang disimpan sudah cukup besar sekitae 4 gigabite.

gambar di generasi kelima. Dan data pad video game yang diggunakan cukup besar. Seperti Sony play station 3 menggunakan blue ray , blue ray berkapasitas 200 giga bite bahkan sekarang sudah ada yang 500 gigabite.Semua console ini juga bisa dimainkan didalam PC atau computer tetapi computer anda harus

berspesifikasi bagus sehingga dapat menampilkan gambar real dengan kualitas gambar yang bagus.

2.1.4 Pembuatan Game

Membuat sebuah game terlebih dahulu pembuat *game* harus membuat deskripsi yang menceritakan *game* yang akan dibuat. Para pengembang *game komersial* biasanya mempersiapkan dokumen-dokumen yang berisi *design game* yang sangat panjang sebelum memulai pembuatan *game*. Namun *design game* yang sederhana dapat membuat pembuatan *game* yang lebih menyenangkan. Dari *design* yang telah dibuat kemudian dapat diketahui semua elemen-elemen berbeda yang dibutuhkan dalam pembuatan *game*, misalnya karakter *user*, karakter musuh, animasi serangan dan sebagainya. Membuat *game* akan membutuhkan gambar dari tiap elemen-elemen yang ada. Membuat *game* akan membutuhkan gambar dari tiap elemen-elemen yang ada, *background image*, dan *sound*. Semua hal diatas dapat dikatakan sebagai *resources game*. (Dawang Muctar,2005)

2.1.5 Klasifikasi Game

Klasifikasi Game Meliputi

- a. *Sprites* : kumpulan gambar-gambar yang digunakan untuk *icon*, karakter ataupun gambar bergerak dalam game. *Sprites* dapat dibuat dari gambar-gambar yang sudah dibuat dalam sebuah *art packages*, *dawnload* dari *internet* maupun dibuat secara manual dengan menggunakan *software* seperti *Adobe Photoshop*.
- b. *Objects* : bagian dari *game* yang mengontrol bagaimana elemen-elemen tersebut bergerak dan berinteraksi satu sama lain. Contohnya karakter dalam *game*.
- c. *Events dan Action* : *event* adalah hal-hal penting yang terjadi didalam sebuah *game*, seperti ketika *object* saling bertumbukan maupun ketika *user* menekan tombol pada *keyboard*, *actions* adalah hal-hal yang terjadi sebagai respon dari sebuah *events*, seperti perubahan arah pergerakan *object*, perubahan score, atau memainkan sebuah lagu.

- d. *Background* : gambar-gambar yang digunakan sebagai gambar latar belakang *game* dan merupakan jenis dari *resource*, seperti *sprites* dan *object*. Pada umumnya ukuran *background* sama dengan *resolusi game*.
- e. *Sound* : juga termasuk *resources game*. Jenis *sound* dalam *game* ada dua, yaitu *music* dan *sound effect*. *Sounds* sangat mempengaruhi *atmosfer* dalam *game*. *Master* berupa lagu yang terus mengalun selama *game* dimainkan, namun lagu yang dimainkan pada umumnya bergantung pada tempat atau kondisi *game*, *sounds effect* adalah suara yang hanya terdengar ketika terjadi suatu hal tertentu, misal penekanan tombol, karakter terkena pukulan dan sebagainya. Kegunaan dari *sound effect* adalah untuk membantu *user* mengetahui *action* yang dilakukan maupun hal yang sedang terjadi. (Dawang Muctar, 2005)
- f. *Status* : sekumpulan informasi yang berguna untuk orang yang memainkan sebuah *game* untuk mengetahui kondisi permainan saat itu. Bergantung dari jenis *game* nya. Informasi yang disampaikan melalui status dapat bermacam-macam. Berikut beberapa macam status pada *game* beserta penjelasannya.:
 - 1) *Score* adalah nilai yang diperoleh oleh orang yang memainkan sebuah *game*. Status ini sangat sering digunakan pada sebuah *game* untuk mengetahui kemampuan yang memainkannya.
 - 2) *High score* adalah nilai tertinggi yang pernah dicapai orang yang memainkan sebuah *game*. Status ini biasanya terdapat pada *game* yang dapat diselesaikan dalam waktu yang cukup singkat, sehingga membuat orang yang memainkannya tertarik untuk mengulangi permainan dari awal, dengan harapan mendapat *high score* yang lebih tinggi.
 - 3) *HP (Hit Point)* adalah status yang menunjukkan jumlah maksimal *damage* yang dapat diterima oleh karakter dalam *game*. Apabila bernilai 0 maka karakter tersebut dinyatakan mati atau kalah. Status ini biasanya terdapat pada *game* ber-genre *fighting*. *Action* atau *RPG (Role Playing Game)*

- 4) *SP (Skill Point)* adalah sejumlah point yang dapat digunakan untuk menggunakan skill. Jumlah yang dibutuhkan bergantung dari *skill* yang digunakan. Status ini biasanya terdapat pada *game ber-genre RPG*.
- 5) *Strength* adalah jumlah kekuatan karakter dalam *game*. Nilai status biasanya menentukan kekuatan serangan karakter. Status ini biasanya terdapat pada *game ber-genre fighting action atau RPG*.
- 6) *Vitality* adalah jumlah pertahanan karakter dalam *game*. Nilai status ini biasanya akan mempengaruhi jumlah HP dan pertahanan yang dimiliki karakter. Status ini biasanya terdapat pada *game ber-genre fighting, action atau RPG*.
- 7) *Speed* adalah kecepatan dalam *game*. Nilai status ini biasanya akan mempengaruhi seberapa cepat karakter akan bergerak atau mendapatkan giliran serta menentukan mudah atau tidaknya terkena serangan musuh.
- 8) *Intelligence* adalah jumlah kekuatan sihir dari karakter dalam *game*. Nilai status ini biasanya akan mempengaruhi jumlah SP, serangan sihir dan ketahanan terhadap serangan sihir. Status ini biasanya terdapat pada *game ber-genre action RPG*.
- 9) *Luck* adalah keberuntungan karakter dalam *game*. Nilai status ini biasanya akan mempengaruhi karakter dalam mendapatkan hal-hal yang sulit didapatkan dalam *game*, seperti terjadinya *critical hit*.
- 10) *Status point* adalah jumlah point yang biasanya digunakan untuk menambah status karakter seperti *strength, vitality, intelligence, speed dan luck*. Status ini biasanya terdapat pada *game ber-genre action atau RPG*.
- 11) *Time* adalah status yang digunakan untuk berbagai macam fungsi dalam *game*, misal untuk menentukan waktu yang tersisa untuk mencapai tujuan tertentu, jumlah waktu yang telah dihabiskan seseorang untuk memainkan *game* tersebut dan lain sebagainya. Status ini biasanya terdapat pada *game ber-genre fighting, action atau RPG*.
- 12) *Experience* dalam dunia *game* adalah sejumlah point yang biasanya digunakan untuk menentukan level karakter dalam *game*. Status ini biasanya terdapat pada *game ber-genre action atau RPG*. (Muctar,2005)

- g. *Turn Based* : sebuah sistem dalam *game* untuk menentukan giliran dimana hanya satu user yang dapat bergerak dalam suatu waktu tertentu. Contoh *game* antara lain *Tic Tac Toe*, *scrabble*, *catur*, permainan kartu dan lain-lain.
- h. sudut pandang *user*, sudut pandang *user* dalam *game* ada 2, yaitu:
 - 1) *First person View* adalah sudut pandang orang pertama, dimana, *user* dijadikan seperti karakter yang dimainkan, sehingga *user* melihat dari sudut pandang karakter yang dimainkan. Contohnya *game counter Strike*, *HALO*
 - 2) *Third person View* adalah sudut pandang orang ketiga, dimana *user* dapat melihat keseluruhan tubuh karakter dalam *game*. Contoh : *Ragnarok Online*, *FATE*
- i. *Elemen dasar* : setiap permainan digital memiliki elemen-elemen dasar berupa:
 - 1) *Grafis*, yakni tampilan *game* tersebut di layar atau *display* alat elektronik, bisa berupa kumpulan *pixel* (satuan unit gambar terkecil atau salah satu bagian gambar tersebut menunjukkan gerakan yang adalah hasil manipulasi si pemain)
 - 2) *Interface*, alat pengantara dimana pemain memanipulasi *game* yang mengentrai dan memproses manipulasi pemain menjadi *input* dalam *game*, dapat berupa *keyboard*, *mouse*, *joystick* ataupun tampilan –tampilan tertentu dalam *game*.
 - 3) Aktivitas pemain, yakni jantung dari pengalaman bermain *game*, tanpa aktivitas pemain maka tidak akan ada suatu permainan; Aktivitas bisa bersifat *diegetic* (bagaimana karakter pemain berubah aktivitas pemain) dan bersifat *extradiegetic* (apa yang terjadi secara fisik pada diri pemain tersebut).
 - 4) Sebuah algoritma, yakni jantung dari program *game* yang merupakan serangkaian program berisi otak dan seperangkat aturan yang mendasari dinamika *game* tersebut.
 - 5) Contoh kasus *game* yang dibuat menggunakan visual basic yaitu *game ular tangga* *game ular tangga* adalah sebuah permainan dimana masing-masing pemain berusaha untuk mencapai kotak akhir.

2.1.6 Animasi

2.1.6.1 Pengertian Animasi

Animasi adalah gambar bergerak berbentuk dari sekumpulan objek (gambar) yang disusun secara beraturan mengikuti alur pergerakan yang telah ditentukan pada setiap pertambahan hitungan waktu yang terjadi. Gambar atau objek yang dimaksud dalam definisi di atas bisa berupa gambar manusia, hewan, maupun tulisan. Pada proses pembuatannya animator harus menggunakan logika berfikir untuk menentukan alur gerak suatu objek dari keadaan awal hingga keadaan akhir objek tersebut. Perencanaan yang matang dalam perumusan alur gerak berdasarkan logika yang tepat akan menghasilkan animasi yang menarik untuk disaksikan. Animasi juga merupakan suatu teknik menampilkan gambar berurut sedemikian rupa sehingga penonton merasakan adanya ilusi gerakan (motion) pada gambar yang ditampilkan. Secara umum ilusi gerakan merupakan perubahan yang dideteksi secara visual oleh mata penonton sehingga tidak harus perubahan yang terjadi merupakan perubahan posisi sebagai makna dari istilah ‘gerakan’. Perubahan seperti perubahan warna juga bisa dikatakan sebuah animasi.

Animasi merupakan salah satu bagian grafika komputer yang menyajikan tampilan-tampilan yang sangat atraktif juga merupakan sekumpulan gambar yang ditampilkan secara berurutan dengan cepat untuk mensimulasi gerakan yang hidup.

2.1.6.2 Macam-Macam Animasi

Berikut adalah macam-macam animasi yang umum digunakan

1. Animasi Cel

Kata cel berasal dari kata “celluloid” yang merupakan materi yang digunakan untuk membuat film gambar bergerak pada tahun - tahun awal animasi. Sekarang material film dibuat dari asetat (acetate). Biasanya digambar dengan menggunakan tangan (hand-drawn animation). Animasi cel biasanya merupakan lembaran - lembaran yang membentuk animasi tunggal.

Masing - masing sel merupakan bagian yang terpisah, misalnya antara obyek dengan latar belakangnya, sehingga dapat saling bergerak mandiri.

2. Animasi Frame

Animasi Frame adalah bentuk animasi paling sederhana. Dalam sebuah film, serangkaian frame bergerak dengan kecepatan minimal 24 frame per detik, agar tidak terjadi jitter.

3. Animasi Sprite

Pada animasi ini, gambar digerakkan dengan latar belakang yang diam. Sprite adalah bagian dari animasi yang bergerak secara mandiri. Dalam animasi ini yang dapat kita edit adalah animasi dari layar yang mengandung sprite, tapi kita tidak dapat mengedit bagian dalam yang ditampilkan oleh layar untuk masing-masing frame seperti pada animasi frame.

4. Animasi Path

Animasi path adalah animasi dari obyek yang bergerak sepanjang garis kurva yang ditentukan sebagai lintasan. Pada kebanyakan dalam animasi path dilakukan juga efek looping yang membuat gerakan path terjadi secara terus menerus.

5. Animasi Spline

Spline adalah representasi matematis dari kurva. Sehingga gerakan obyek tidak hanya mengikuti garis lurus, melainkan berbentuk kurva.

6. Animasi Vektor

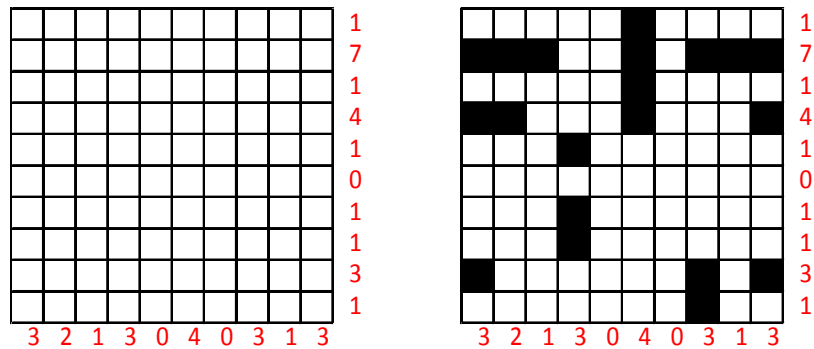
Vektor adalah garis yang memiliki ujung pangkal, arah dan panjang. Animasi vektor mirip dengan animasi sprite, tapi animasi sprite menggunakan bitmap sedangkan animasi vektor menggunakan rumus matematika untuk menggambarkan sprite-nya.

7. Animasi Character

Animasi character biasanya terdapat di film kartun. Semua bagian dalam film kartun selalu bergerak bersamaan. Software yang biasa digunakan adalah Maya Unlimited. Contoh film kartun yang dibuat dengan Maya Unlimited adalah Toy Story dan Monster

2.2 Battleship

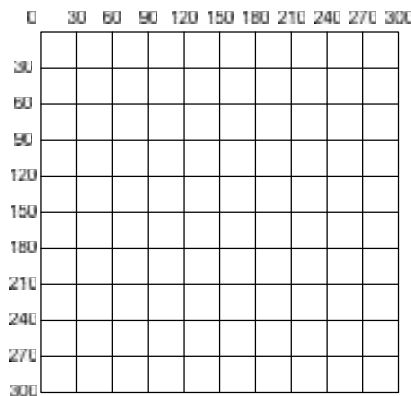
Permainan ini menyediakan beberapa buah tempat yang disusun oleh kotak-kotak sebagai medan perang (Bradley 1990).



Gambar 2.1.Medan perang pada permainan *Battleship*

Kapal-kapal perang dengan ukuran yang berbeda-beda disusun dalam medan perang masing-masing. Letak dari kapal-kapal perang ini tidak terlihat dalam komputer pemain lawan. Setiap pemain berusaha untuk menghancurkan kapal-kapal perang pemain lawan dengan cara meng-klik kotak-kotak yang dianggap sebagai letak dari kapal-kapal perang lawan. Setiap pemain hanya boleh menebak satu kali saja setiap gilirannya. Pemain yang masih menyisakan kapal perang yang memenangkan permainan (*game*).

Medan perang dirancang dengan ukuran 10 kotak x 10 kotak. Kotak-kotak ini digambarkan dengan dibatasi oleh garis-garis. Jarak garis-garis tersebut sebanyak 30 *twips* baik secara horizontal maupun vertikal. Sketsa dari gambar medan perang tersebut dapat dilihat pada gambar berikut ini :



Gambar 2.2 Sketsa gambar dari medan perang

Kotak-kotak tersebut akan disimpan dalam bentuk *array* zone 2 (x,y) dengan cara penomoran kotak tersebut adalah sebagai berikut :

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Gambar 2.3 Penomoran kotak pada medan perang

Penentuan Posisi Kapal akan dilakukan dengan cara kapal yang diletakkan akan ditentukan posisinya dan disimpan ke dalam sebuah variabel *Array* yang memiliki dua buah properti yaitu X untuk menyimpan posisi kolom dari kapal, Y untuk menyimpan posisi baris dari kapal.

Pengecekan Posisi Tembakan dan Kapal dilakukan dengan cara jika posisi (X,Y) yang diklik sama dengan salah satu dari posisi kapal yang disimpan dalam *array* tersebut maka berarti kapal tertembak dan posisi bagian kapal yang tertembak tersebut ditandai. Posisi koordinat yang ditembak juga ditandai. Proses perhitungan posisi x dan y tersebut sama dengan proses perhitungan pada penentuan posisi kapal di atas.

Teka-teki *Battleship* adalah salah satu teka-teki yang membutuhkan logika sebagai dasar penyelesaiannya. Teka-teki ini direpresentasikan dengan kisi yang berisi sekumpulan kotak-kotak persegi dengan angka-angka di pinggir kanan dan bawah kisi. Tujuan dari teka-teki ini adalah menebak letak posisi kapal-kapal yang tersembunyi di dalam teka-teki *Battleship* tersebut. Satu-satunya petunjuk adalah angka-angka di pinggir kisi yang menunjukkan jumlah bagian kapal yang tersembunyi di dalam setiap kolom ataupun baris yang bersangkutan serta beberapa bagian kapal ataupun bagian air pada posisi tertentu yang bervariasi pun dapat ditunjukkan dari awal teka-teki. Setiap teka-teki *Battleship* ini memiliki solusi yang unik. Sehingga, teka-teki ini pun dapat diselesaikan dengan mudah

dengan bantuan algoritma. Di dalam sistem ini akan dibahas bagaimana cara menemukan solusi dari teka-teki *Battleship* dengan menggunakan penerapan algoritma Runut-Balik, yang berbasis pada algoritma *Depth First Search* (DFS).

2.3 Visual basic 6.0

Visual basic adalah salah satu bahasa pemrograman komputer yang digunakan untuk aplikasi windows yang berbasis (GUI Graphical User Interface). Visual basic merupakan *event-driven programming* (Pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa event/kejadian tertentu (tombol diklik, menu dipilih, dan lainlain). Ketika event terdeteksi, kode yang berhubungan dengan event (prosedur event) akan dijalankan. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (Beginner s All-Purpose Symbolic Instruction Code) yang dikembangkan pada era 1950-an. Visual Basic merupakan salah satu Development Tool yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (Object Oriented Programming = OOP).

Visual basic ini banyak dimanfaatkan untuk pembuatan aplikasi yang berbasis desktop. Selain itu biasa pemrograman visual basic bisa juga digunakan untuk pembuatan game seperti, game *tic-tac*, ular tangga, dan game tebak kata. Berikut salah satu game yang dibuat dengan bahasa pemrograman *visual basic*:

2.3.1 Game Ular Tangga

permainan ular tangga merupakan satu set alat bermain yang berupa sebuah papan atau karton bergambar kotak-kotak biasanya berukuran 10×10 kotak. Tiap kotak diberi nomor urut mulai dari nomor 1 dari sudut kiri bawah sampai nomor 10 disudut kanan bawah, lalu dari kanan ke kiri mulai nomor 11 baris kedua sampai nomor 20 dan seterusnya sampai nomor 100 di sudut kiri atas. Ciri khas yang tidak pernah berubah adalah kehadiran ular dan tangga dalam gambar permainan tersebut. Giliran bermain dan jumlah langkah yang akan

dimainkan ditentukan menggunakan lemparan dadu. Menurut Wulandari, permainan ular tangga dapat digunakan untuk mencapai tujuan kognitif secara umum, terutama yang termasuk pengenalan, membedakan, pemahaman, hafalan, rumus-rumus matematika, dan lain-lain. Sedangkan tujuan afektif dapat ditinjau dari adanya kesatuan kerja sama, bekerja sendiri, tambah percaya diri, tidak putus asa dan pantang menyerah. Sementara tujuan psikomotor dari permainan ular tangga ini kurang berperan. Selain itu permainan ini juga menambah motivasi pada pokok-pokok bahasan yang biasanya kurang menarik perhatian siswa, misalnya pemahaman konsep yang abstrak, hafalan, cerita, dan lain-lain.

2.4 Perangkat Lunak Berorientasi Objek

2.4.1 OOA (*Object Oriented Analysis*)

OOA mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi objek. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan (*requirement*) yang diperoleh dari semua pihak yang berkepentingan. (Mis: klien, developer, pakar, dll). Dokumen permintaan memiliki 2 fungsi yaitu : memformulasikan kebutuhan klien dan membuat suatu daftar tugas

Analisis berorientasi obyek (OOA) melihat pada domain masalah, dengan tujuan untuk memproduksi sebuah model konseptual informasi yang ada di daerah yang sedang dianalisis. Model analisis tidak mempertimbangkan kendala-kendala pelaksanaan apapun yang mungkin ada, seperti konkurensi, distribusi, ketekunan, atau bagaimana sistem harus dibangun. Kendala pelaksanaan ditangani selama desain berorientasi objek (OOD). Analisis dilakukan sebelum Design.

Sumber-sumber untuk analisis dapat persyaratan tertulis pernyataan, dokumen visi yang formal, wawancara dengan *stakeholder* atau pihak yang berkepentingan lainnya. Sebuah sistem dapat dibagi menjadi beberapa domain, yang mewakili bisnis yang berbeda, teknologi, atau bidang yang diminati, masing-masing dianalisis secara terpisah.

Hasil analisis berorientasi objek adalah deskripsi dari *apa* sistem secara fungsional diperlukan untuk melakukan, dalam bentuk sebuah model konseptual. Itu biasanya akan disajikan sebagai seperangkat menggunakan kasus, satu atau lebih UML diagram kelas, dan sejumlah diagram interaksi. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan pelanggan.

2.4.2 OOD (*Object Oriented Design*)

Object-oriented design adalah metoda untuk meng-arahkan arsitektur perangkat lunak yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Model kebutuhan-kebutuhan yang dibuat pada fase analisis diperkaya dalam fase perancangan. Kadang-kadang ditambahkan lebih banyak lagi atribut dan pelayanan dan ditambahkan antarmuka obyek-obyek.

Object Oriented Design bertujuan untuk mengoptimalkan *maintainability*, *reusability*, *enhanceability* dan *reliability*.

Memberikan blueprint untuk implementasi dan menspesifikasi “HOW” dengan menspesifikasi *class definitions*, *class categories*, *subsystems*, *system architectures*. *Object Oriented Design* merupakan *Object-Oriented Analysis*

OOD mengubah model konseptual yang dihasilkan dalam analisis berorientasi objek memperhitungkan kendala yang dipaksakan oleh arsitektur yang dipilih dan setiap non-fungsional-teknologi atau lingkungan – kendala, seperti transaksi throughput, response time, run – waktu platform, lingkungan pengembangan, atau bahasa pemrograman.

2.5 UML (*Unified Modelling Language*)

Menurut Munawar, (2005) *Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

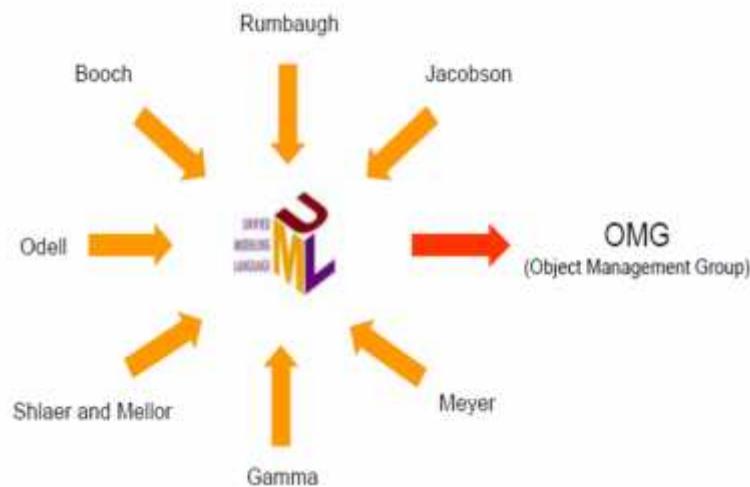
Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak.

Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). (Julius Hermawan, 2005).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah:

1. Metodologi booch
2. Metodologi coad
3. Metodologi OOSE
4. Metodologi OMT
5. Metodologi shlaer-mellor
6. Metodologi wirfs-brock

Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan *group* atau perusahaan lain yang menggunakan metodologi yang berlainan.



Gambar 2.4 Beberapa Contoh Pemodelan Berorientasi Objek

2.5.1 Konsepsi Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar dibawah.

Major Area	View	Diagrams	Main Concepts
Structural	Static View	Class Diagram	Class, Association, Generalization, Dependency, Realization, Interface
	Use Case View	Use Case Diagram	Use Case, Actor, Association, Extend, Include, Use Case Generalization
	Implementation View	Component Diagram	Component, Interface, Dependency, Realization
	Deployment View	Deployment Diagram	Node, Component, Dependency, Location
Dynamic	State Machine View	Statechart Diagram	State, Event, Transition, Action
	Activity View	Activity Diagram	State, Activity, Completion Transition, Fork, Join
	Interacion View	Sequence Diagram Collaboration Diagram	Interaction, Object, Message, Activation Collaboration, Interaction, Collaboration, Role, Message
Model Management	Model Management View	Class Diagram	Package, Subsystem, Model
Extensibility	All	All	Constraint, Stereotype, Tagged Values

Gambar 2.5 Konsep Dasar UML

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bias kita pahami dengan mudah apabila kita melihat gambar diatas dari *diagram*. *Main consepts* bisa kita pandang sebagai tern yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagram tersebut.

a. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal.

Sebuah *Use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Artifact didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. UML (Astah) adalah model perancangan sistem berorientasi objek, model perancangan sistem ini membahas model perancangan sebagai berikut :



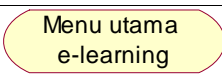


1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*

b. *Activity Diagram*

Pada dasarnya. Diagram aktivitas adalah Diagram *flowchart* yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain. Kegunaan diagram ini adalah untuk memodelkan *workflow* atau jalur kerja,

memodelkan operasi, bagaimana objek-objek bekerja, aksi-aksi dan pengaruh terhadap objek. Simbol-simbol yang terdapat dalam *Activity Diagram*, sebagai berikut :

Tabel 2.1 Simbol *Activity Diagram*

Keterangan	Simbol
Titik Awal atau permulaan.	
Titik Akhir atau akhir dari aktivitas.	
<i>Activity</i> , atau aktivitas yang dilakukan oleh aktor.	
<i>Decision</i> , atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

Activity diagram merupakan salah satu diagram yang umum digunakan dalam *UML* untuk menjabarkan proses atau aktivitas dari aktor. Sebagai contoh, pelanggan melakukan *login* (masuk) pada halaman *website* untuk bergabung, jika pelanggan belum terdaftar, maka akan ditolak oleh sistem dan dikembalikan.

Di dalam *Activity diagram* tersebut dijelaskan bahwa *user* melakukan proses *login* untuk dapat memasuki area sistem, jika proses *login* dan/atau *user* belum teregistrasi, maka *user* akan ditolak oleh sistem tersebut dan diberi pesan *error*. Selain itu, bila *user* telah teregistrasi dan memasukkan kode *login* dengan benar maka akan diberi akses untuk masuk ke sistem, dan diberikan pesan sukses. *User* dapat *logout* (keluar) untuk mengakhiri sesi.

c. *Sequence diagram*

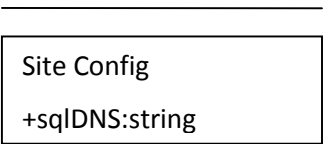
Sequence diagram mendokumentasikan komunikasi/interaksi antar kelas-kelas. Diagram ini menunjukkan sejumlah obyek dan *message* (pesan) – yang diletakkan diantara obyek-obyek didalam *use case*. Perlu diingat bahwa di dalam diagram ini, kelas-kelas dan aktoraktor diletakkan dibagian atas diagram dengan

urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara vertikal terhadap kelas dan aktor.

b. *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Berikut adalah notasi – notasi yang ada pada class diagram :

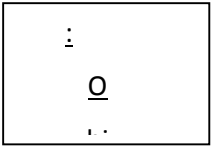


Tabel 2.2 Notasi pada *Class Diagram*

<i>Nama</i>	<i>Keterangan</i>	<i>Notasi</i>
<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhirmendefinisikan method-method dari sebuah <i>class</i> .	

c. *Collaboration Diagram*

Collaboration diagram menggunakan prinsip yang sama dengan *sequence diagram*, sama-sama memodelkan interaksi antar obyek-obyek, yang membedakannya hanya cara penggambarannya saja. Pada collaboration diagram ini, obyek-obyek dan *message* (pesan) yang ada digambarkan mirip seperti flowchart, hanya saja, untuk menjaga urutan pesan yang diterima oleh masing-masing obyek, pesan-pesan tersebut diberi nomor urutan pesan. Berikut adalah notasi untuk *collaboration diagram*

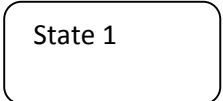
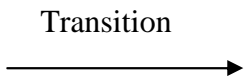


Tabel 2.3 Notasi *collaboration* diagram

<i>Name</i>	<i>Keterangan</i>	Notasi
<i>Object</i>	<i>Object</i> merupakan instance dari sebuah class. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan object, maka actor juga dapat disertakan ke dalam collaboration diagram. Simbol Actor sama dengan simbol pada <i>Actor Use Case Diagram</i> .	
<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar obyek.	

d. *State Diagram*

Statechart diagram, atau yang biasa juga disebut *state diagram* digunakan untuk mendokumentasikan beragam kondisi/keadaan yang bisa terjadi terhadap sebuah *class* dan kegiatan apa saja yang dapat merubah kondisi/keadaan tersebut. Contohnya sebuah televisi yang dapat berada dalam kondisi menyala atau mati, jika tombol “power” ditekan maka televisi akan menyala, begitu juga sebaliknya akan mati jika tombol “power” ditekan kembali. Maka disini kita mempunyai sebuah kelas yaitu televisi, 2 *state* yaitu *menyala* dan *mati* dan 2 *transition* yaitu *menyalakan TV* dan *mematikan TV*. Tidak seperti diagram-diagram *behavioural* lainnya yang memodelkan interaksi diantara beberapa *class*, *state diagram* justru biasanya hanya memodelkan transisi yang terjadi hanya pada sebuah *class*. Berikut adalah notasi *state diagram* :

Tabel 2.4 Notasi *Statechart* Diagram

<i>Nama</i>	Keterangan	<i>Notasi</i>
<i>State</i>	Notasi <i>State</i> menggambarkan kondisi sebuah entitas, dan digambarkan dengan segiempat yang pinggirnya tumpul dengan nama state didalamnya.	
<i>Transition</i>	Sebuah <i>Transition</i> menggambarkan sebuah perubahan kondisi objek yang disebabkan oleh sebuah event. <i>Transition</i> digambarkan dengan sebuah anak panah dengan nama event yang ditulis di atasnya, dibawahnya atau sepanjang anak panah tersebut.	
<i>Initialstate</i>	<i>Initial State</i> adalah sebuah kondisi awal sebuah object sebelum ada perubahan keadaan. <i>Initial State</i> digambarkan dengan sebuah lingkaran solid. Hanya satu <i>Initial State</i> yang diizinkan dalam sebuah diagram	
<i>FinalState</i>	<i>Final State</i> menggambarkan ketika objek berhenti memberi respon terhadap sebuah event. <i>Final State</i> digambarkan dengan lingkaran solid didalam sebuah lingkaran kosong.	

e. *Component Diagram*

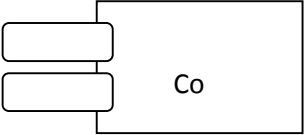
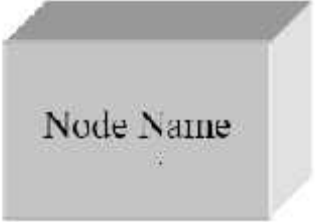

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi software dari sebuah class. Komponen bias berupa tabel, *file data*, *file exe*, file DLL, dokumen dan lain-lain.

Component diagram mengandung komponen, *interface* dan *relationship*. Komponen diagram ini digunakan pada saat anda ingin memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Secara umum dapat kita katakan bahwa component diagram kita gunakan untuk menjelaskan kebergantungan antar beragam komponen-komponen *software*

seperti misalnya kebergantungan antara *file-file executable* dengan *file-file sumbernya (source file)* dan lain-lain.

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. Berikut adalah notasi-notasi yang digunakan pada *deployment diagram*.

Tabel 2.5 Notasi *Deployment Diagram*

<i>Nama</i>	Keterangan	<i>Notasi</i>
<i>Component</i>	Pada deployment diagram, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.	
<i>Node</i>	Node menggambarkan bagian bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.	
<i>Association</i>	Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen hardware.	

2.6 Jaringan Komputer

Sistem jaringan komputer adalah gabungan atau kumpulan dari beberapa komputer yang dapat diakses secara bersama-sama (seperti *floopy disk*, CD-ROM, *printer*, dan sebagainya), dan dapat berhubungan dengan komputer induk sistem lainnya yang letaknya berjauhan.

Komunikasi data dapat diartikan pengiriman data secara elektronik dari satu tempat ke tempat lain melalui suatu media komunikasi, dan data yang dikirimkan tersebut merupakan hasil atau akan diproses oleh suatu sistem komputer (Husni, 2003).

Sistem jaringan dapat memiliki peralatan pada komputer *server* untuk dipakai secara bersama dengan komputer *client*-nya. Namun pada setiap komputer lokal dapat juga dipasang peralatan khusus untuk keperluan komputer lokal tersebut.

Beberapa komponen dasar yang biasanya membentuk suatu LAN adalah sebagai berikut (Husni, 2003).

1. *Host* atau *node*, yaitu sistem komputer yang berfungsi sebagai sumber atau penerima dari data yang dikirimkan. *Node* ini dapat berupa:
 - a. *Server* : komputer tempat penyimpanan data dan program-program aplikasi yang digunakan dalam jaringan,
 - b. *Client* : komputer yang dapat mengakses sumber daya (berupa data dan program aplikasi) yang ada pada server,
 - c. *Shared peripheral* : peralatan-peralatan yang terhubung dan digunakan dalam jaringan (misalnya, printer, scanner, harddisk, modem, dan lain-lain).
2. *Link*, adalah media komunikasi yang menghubungkan antara node yang satu dengan node lainnya. Media ini dapat berupa saluran transmisi kabel dan tanpa kabel.
3. *Software* (Perangkat Lunak), yaitu program yang mengatur dan mengelola jaringan secara keseluruhan. Termasuk di dalamnya sistem operasi jaringan yang berfungsi sebagai pengatur komunikasi data dan periferal dalam jaringan.

2.7 Lokal Area Network (LAN)

Menurut Jogiyanto (2004) “ LAN adalah suatu network yang terbatas dalam jarak atau area setempat. Network ini banyak digunakan satu perusahaan yang menghubungkan antara departemen-departemen dalam satu gedung”.

LAN juga dapat didefinisikan sebagai network atau jaringan sejumlah system computer yang lokasinya terbatas dalam satu gedung, satu kompleks gedung atau suatu kampus dan tidak menggunakan media fasilitas komunikasi umum seperti telepon, melainkan pemilik dan pengelola media komunikasinya adalah pemilik LAN itu sendiri.

Lokal Area Network (LAN) merupakan jaringan milik pribadi didalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumber daya (*resource*, misalnya printer) dan saling bertukar informasi. Secara garis besar terdapat dua tipe jaringan LAN, yaitu jaringan *per to per* dan jaringan *Client-Server*. Pada jaringan *per to per*, setiap Komputer yang terhubung ke jaringan dapat bertindak baik sebagai *workstation* maupun server. Sedangkan pada jaringan *client-server*, hanya satu komputer yang bertugas sebagai server atau operator dan komputer lain berperan sebagai *workstation*. LAN seringkali menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai 10 sampai 100 Mbps (megabit/detik) dengan delay rendah (puluhan *mikro second*) dan mempunyai faktor kesalahan yang kecil.

2.8 Arsitektur Jaringan Komputer

Dalam pengimplementasian *Battleship Multi Player Game Strategy* ini adalah menggunakan jaringan *peer to peer*. Ada tiga bentuk arsitektur yang umum digunakan dalam jaringan komputer yaitu: (Husni, 2003).

2.8.1 Jaringan *peer to peer*

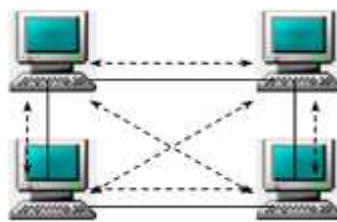
Pada jaringan *peer to peer* (Gambar 2.6), semua komputer memiliki posisi setara/sejajar, dalam hierarki yang sama. Setiap komputer dapat menjadi klien terhadap komputer *peer* lainnya, setiap komputer dapat pula berbagi sumber daya

dengan komputer yang berada dalam jaringan peer-to-peer ini. Sumber daya diletakkan secara desentralisasi pada setiap anggota jaringan, dan tidak memerlukan administrator jaringan.

Aliran informasi bisa mengalir di antara dua komputer secara langsung, di mana pun. Namun, jaringan ini tidak sepenuhnya bebas tanpa kontrol, masih bisa digunakan *password* untuk memproteksi file dan folder, dapat juga diatur agar orang-orang tertentu tidak bisa menggunakan periferal tertentu.

Karena kemudahan pemasangan, pemeliharaan, serta biaya, jaringan ini lebih populer untuk jaringan dengan jumlah komputer yang sedikit (sekitar 2 sampai 20 komputer).

Sifat jaringan *peer to peer* digunakan untuk hubungan antara setiap komputer yang terhubung dalam jaringan komputer yang ada, sehingga komunikasi data terjadi antar komputer dengan hierarki yang sama karena setiap komputer dapat berfungsi sebagai server maupun klien.



Gambar 2.6 Jaringan *peer-to-peer*

2.8.2 IP Address

IP atau *Internet Protocol* mendefinikan bagaimana informasi dilewatkan antara satu sistem dengan sistem yang lain di Internet. *IP Address* atau alamat IP adalah suatu deretan bilangan unik yang mengidentifikasi suatu *host* atau komputer di Internet. Bilangan-bilangan tersebut biasanya ditampilkan dalam kelompok-kelompok yang dipisahkan oleh titik, seperti 202.145.34.57. semua sumber daya di Internet harus mempunyai *IP Address* atau sama sekali tidak dapat hadir di Internet. (Husni, 2003).

Lebih penting dari itu semua, suatu komputer atau peralatan jaringan lain tidak akan pernah dapat terkoneksi atau menjadi bagian suatu jaringan jika tidak memiliki *IP Address*. Artinya setiap komputer harus mempunyai *IP Address*

meskipun hanya terkoneksi kedalam jaringan lokal dan tidak terhubung kedalam jaringan public atau internet. *Network Administrator* juga harus memastikan bahwa tidak terdapat dua atau lebih komputer yang memiliki *IP Address* yang sama.

Suatu *IP Address* terdiri dari 32 *bit* yang bekerja dalam suatu urutan waktu. Dari 32 *bit* kemudian dipecah menjadi 8 *bit* atau lebih dikenal dengan oktet dimana masing-masing oktet dikonversikan dalam bilangan desimal yang berkisar antara 0 sampai 255.

IP Address terbagi menjadi 5 kelas yaitu sebagai berikut: (Husni, 2003).

1. Kelas A

Alamat kelas A adalah:

- | | |
|-----------------------|--|
| 0.0.0.0 | : Dicalangkan, tidak boleh digunakan. |
| 1.0.0.0 s.d 126.0.0.0 | : Tersedia dan boleh digunakan. |
| 127.0.0.0 | : Dicalangkan, digunakan oleh <i>localhost</i> untuk keperluan <i>loopback</i> . |

2. Kelas B

Alamat kelas B adalah:

- | | |
|---------------------------|---------------------------------|
| 128.0.0.0 s.d 191.254.0.0 | : Tersedia dan boleh digunakan. |
| 191.255.0.0 | : Dicalangkan, tidak dipakai. |

3. Kelas C

Alamat kelas C adalah:

- | | |
|-----------------------------|---------------------------------|
| 192.0.0.0 | : Dicalangkan, tidak dipakai. |
| 192.0.1.0 s.d 223.255.254.0 | : Tersedia dan boleh digunakan. |
| 233.255.255.0 | : Dicalangkan. |

4. Kelas D

Alamat kelas D adalah:

- | | |
|-------------------------------|-----------------------------|
| 224.0.0.0 s.d 239.255.255.255 | : Kelompok <i>Multicast</i> |
|-------------------------------|-----------------------------|

5. Kelas E

Alamat kelas E adalah:

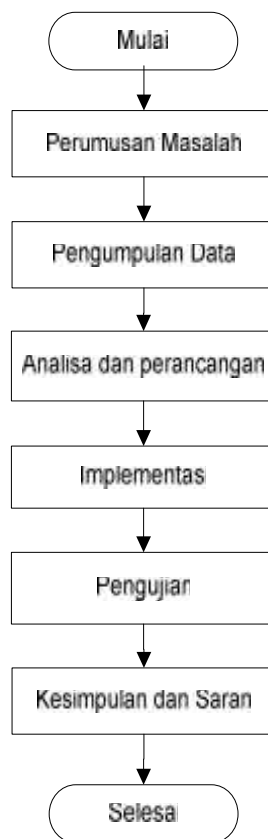
- | | |
|-------------------------------|-----------------------------------|
| 224.0.0.0 s.d 239.255.255.255 | : Dicalangkan (ilegal) |
| 255.255.255.255 | : Dicalangkan, <i>broadcast</i> . |

BAB III

METODOLOGI PENELITIAN

3.1. *Flowchart* Metodologi Penelitian

Flowchart yang digunakan dalam menyelesaikan penelitian tugas akhir ini adalah sebagai berikut:



Gambar 3.1 Tahapan Metodologi Penelitian

Gambar 3.1 di atas merupakan metodologi penelitian yang akan dilakukan dalam penyusunan tugas akhir ini. Dari gambar di atas, dapat diketahui bahwa ada tujuh tahapan yang akan dilakukan untuk menyelesaikan kasus pada penelitian tugas akhir ini yang identifikasi masalah, perumusan masalah, pengumpulan data,

analisa dan perancangan, implementasi, pengujian aplikasi, dan kesimpulan dan saran. Ketujuh tahapan diatas akan dijelaskan pada uraian dibawah ini.

3.2. Perumusan Masalah

Perumusan masalah merupakan tahapan untuk merumuskan masalah setelah dilakukan tahapan identifikasi masalah. Setelah identifikasi masalah selesai, maka tahapan selanjutnya adalah perumusan masalah tentang penelitian yang dilakukan.

3.3. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan dalam pembuatan aplikasi. Pengumpulan data ini dapat diperoleh dari artikel-artikel, jurnal, buku dan juga dari internet yang berhubungan untuk perancangan aplikasi ini.

3.3.1 Jenis Data

Penyusunan laporan skripsi ini, penulis menggunakan data-data sebagai berikut:

a. Data Primer

Data yang dikumpulkan secara langsung, pada pihak atau orang yang mengerti dengan *Visual Basic*. dan seorang personal yang pernah memainkan game battleship ini.

b. Data Sekunder

Data yang diperoleh dalam bentuk yang sudah jadi berupa publikasi dan telah diolah dan dikumpulkan oleh pihak lain. Data sekunder ini antara lain dengan studi pustaka, yaitu data yang bersifat teori yang diperoleh dari berbagai studi pustaka, buku-buku referensi, serta sumber data lainnya yang berhubungan dengan penelitian ini.

3.3.2 Teknik Pengumpulan Data

1. wawancara.

- a. pembuat game

wawancara ini dilakukan kepada orang pembuat game bagaimana cara membuat game apa-apa saja yang diperlukan untuk membuat game.

- b. Pemain game

Wawancara ini dilakukan di pusat permainan game, dalam seminggu saya mewawacarai beberapa orang, game apa yang sering dimainkan dan game apa yang perlu dikembangkan.

2. Studi pustaka

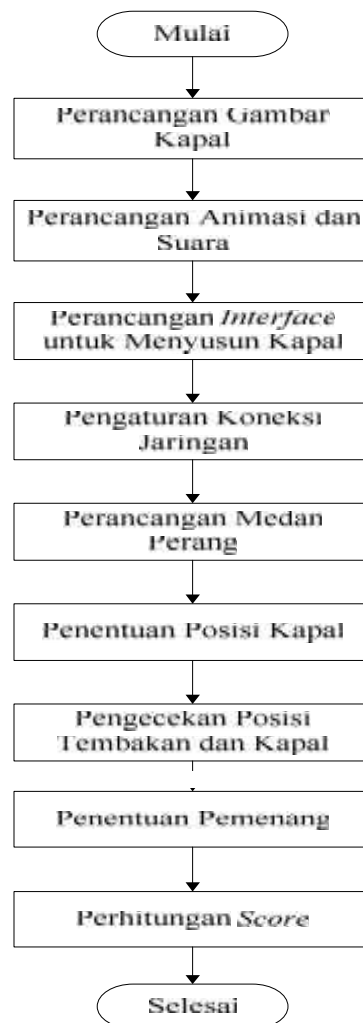
Teknik pengumpulan data dengan cara membaca buku-buku yang berkaitan dengan permasalahan yang penulis bahas guna menambah referensi bagi penulis.

3.4. Analisa dan Perancangan

Analisa dan perancangan merupakan metode yang dilakukan setelah pengumpulan terhadap data-data atau informasi mengenai kasus yang diangkat pada penelitian tugas akhir ini. Analisa berarti metode yang khusus untuk menganalisis masalah yang dapat dimulai dari analisa terhadap alur-alur proses pembuatan aplikasi *battleship game*, kemudian menganalisa model hingga rancang bangun aplikasi itu sendiri. Game strategi *Battleship* pada jaringan ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan didukung oleh beberapa aplikasi lainnya seperti *Adobe Photoshop* dalam perancangan gambar dan *Recorder* dalam perancangan suara. Untuk menghubungkan antara client dengan server yaitu menggunakan *winsock* dengan cara apabila menjadi clien maka masukan IP lawanya. Masukkan Alamat Ip Adddres untuk Setting LAN dengan angka 192.1.1.2 dan subnet mask 255.255.255.0 dan untuk default gateway sama dengan ip addres. Di komputer 2 masukan angka yang sama dengan yang di atas cuma di ganti menjadi 3, misalnya 192.1.1.3. dan IP addres ini juga bisa dirubah atau diganti.

Tahapan analisa dan perancangan ini meliputi beberapa hal yaitu :

1. Perancangn arsitektur aplikasi aplikasi
2. Menentukan objek-objek dan menentukan suara yang digunakan.
3. Merancang *interface*.
4. Pengaturan koneksi jaringan komputer.
5. Penghitungan *score*.
6. Perancangan tampilan.



Gambar 3.2 Flowchart Pembuatan Aplikasi *Battleship*

3.5. Implementasi dan Pengujian

Implementasi dan pengujian merupakan metode terakhir yang digunakan setelah analisa dan perancangan rancang bangun aplikasi selesai dilakukan. Metode ini akan menjelaskan tentang penerapan jalannya rancang bangun aplikasi yang telah dianalisa dan dirancang. Aplikasi tersebut selanjutnya diimplementasikan dan dilakukan pengujian untuk mengetahui tingkat keberhasilan aplikasi yang telah dirancang. Bahasa pemrograman yang digunakan untuk membangun aplikasi ini adalah *Visual Basic 6.0*. selanjutnya dilakukan pengujian.

Implementasi pengembangan aplikasi ini akan dikembangkan pada spesifikasi *hardware* dan *software* berikut:

1. Perangkat keras

Processor : *Pentium(R) Dual-Core CPU T4400 @ 2.20GHz*

Memori (RAM) : 1.00 GB

2. Perangkat Lunak

Sistem operasi : *Windows 7 Profesional 32-bit Operating System*

Bahasa pemrograman : *Visual Basic 6.0*

Sementara untuk tahapan pengujian yang akan dilakukan pada aplikasi *battleship game* Pengujian *blackbox* dan *User Acceptance Test* untuk pengujian tingkah laku aplikasi yang telah dirancang.

3.6. Kesimpulan dan Saran

Tahapan kesimpulan dan saran merupakan akhir dari penelitian tugas akhir ini. Tahapan ini berisi tentang kesimpulan dari hasil-hasil penelitian dan pengujian yang telah dilakukan pada penelitian tugas akhir ini yaitu “Rancang Bangun Aplikasi *Game Battleship Multi Player* pada Jaringan LAN” dan berisi saran-saran yang bersifat membangun serta dapat dijadikan bahan penelitian ulang untuk meneliti dan merancang aplikasi serupa yang lebih baik.

BAB IV

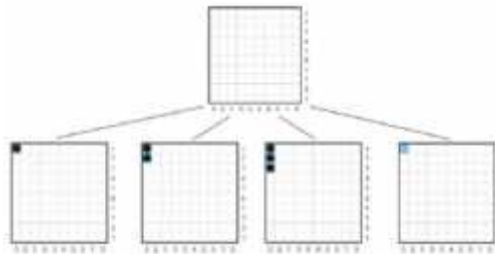
ANALISA DAN PERANCANGAN

Pemahaman secara detail mengenai persyaratan perangkat lunak sangat penting bagi keberhasilan usaha pengembangan perangkat lunak. Tidak peduli bagaimana perangkat lunak di rancang atau dikodekan, program yang dianalisis dan ditentukan secara tidak baik akan mengecewakan pemakainya dan akan membawa kegagalan bagi pengembangannya. Model-model data yang dibutuhkan, aliran kontrol dan informasi, dan tingkah laku operasional diciptakan. Pemecahan alternatif dianalisis dan dialokasikan ke berbagai elemen perangkat lunak. (Pressman,1997)

4.1. Analisa Sistem Lama

Pada penelitian Sumarsono (2009) solusi dari teka-teki *Battleship* dapat ditemukan dengan berbasiskan algoritma Runut-Balik. Fungsi pembangkit pada kasus ini adalah semua kemungkinan bagian kapal baik secara vertikal maupun horizontal serta bagian air. Fungsi pembatas yang digunakan adalah perbandingan jumlah bagian kapal pada suatu kolom dan baris dengan angka pembanding yang berada di pinggir kanan (untuk baris) dan bawah (untuk kolom). Selain itu, fungsi pembatas lainnya adalah pengecekan ada atau tidaknya bagian kapal di sekitar lokasi penempatan bagian kapal. Apabila ada, maka lokasi tersebut tidak dapat ditempatkan bagian kapal, sehingga dapat ditempatkan bagian air. Algoritma pencarian solusi teka-teki ini dapat dimulai dengan mengisi kotak pada baris pertama dan kolom pertama dengan bagian kapal yang memenuhi fungsi pembatas. Misalnya, pada gambar 4.1, pada baris pertama serta kolom pertama, komponen yang dapat dibangkitkan adalah komponen yang memenuhi angka pinggir kanan baris tersebut yaitu satu dan pinggir bawah kolom tersebut yaitu tiga. Komponen yang memenuhi kriteria tersebut adalah bagian air, kapal *submarine* (1 bagian), kapal *destroyer* (2 bagian) secara vertikal, dan kapal

(3 bagian) secara vertikal. Sehingga kita dapat memperoleh pohon ruang status awal seperti pada gambar 4.1 dibawah ini.

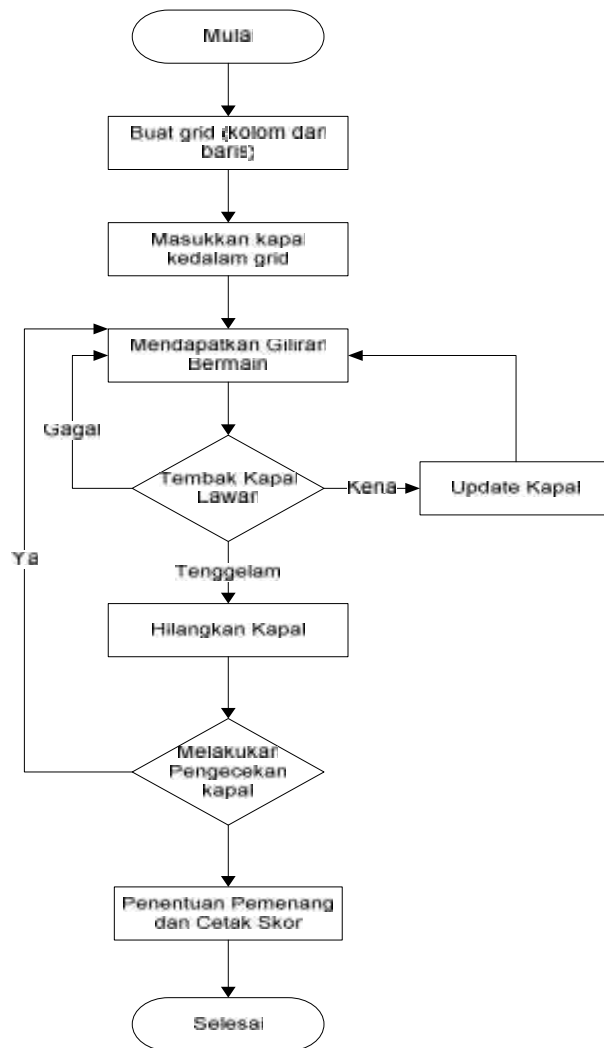


Gambar 4.1 Pohon ruang status awal

Lalu, kita ambil simpul pertama, yang merupakan kapal *submarine*, untuk kemudian dijadikan simpul-E. Dari status tersebut, kita dapat kembangkan lagi dengan mengisi kotak pada baris yang sama dan kolom selanjutnya. Hal ini dilakukan terus menerus, hingga mencapai kolom terakhir pada baris tersebut. Di sini setelah kotak tersebut diisi dengan komponen yang memenuhi fungsi pembatas, dilakukan proses pengecekan apakah jumlah bagian kapal pada baris tersebut sudah sama dengan angka pinggir kanan pada baris yang bersangkutan. Apabila tidak sama, simpul dengan status tersebut dimatikan, lalu dilakukan proses runut-balik menuju simpul orangtua di atasnya yang masih hidup.

Simpul orangtua tersebut pun kemudian menjadi simpul-E. Begitu seterusnya hingga mencapai kolom terakhir. Pada baris terakhir, setelah proses pengisian pada kotak yang sedang ditunjuk, dilakukan pengecekan apakah bagian kapal pada kolom yang ditunjuk sudah sesuai dengan angka pinggir bawah pada kolom yang bersangkutan. Apabila sudah sama, maka proses yang sama dilakukan di kolom selanjutnya. Namun apabila belum memenuhi, maka dilakukan proses runut-balik seperti sebelumnya hingga dapat memenuhi angka pinggir bawah pada kolom yang bersangkutan.

Flowchart permainan *battleship* dapat dilihat pada gambar dibawah ini.

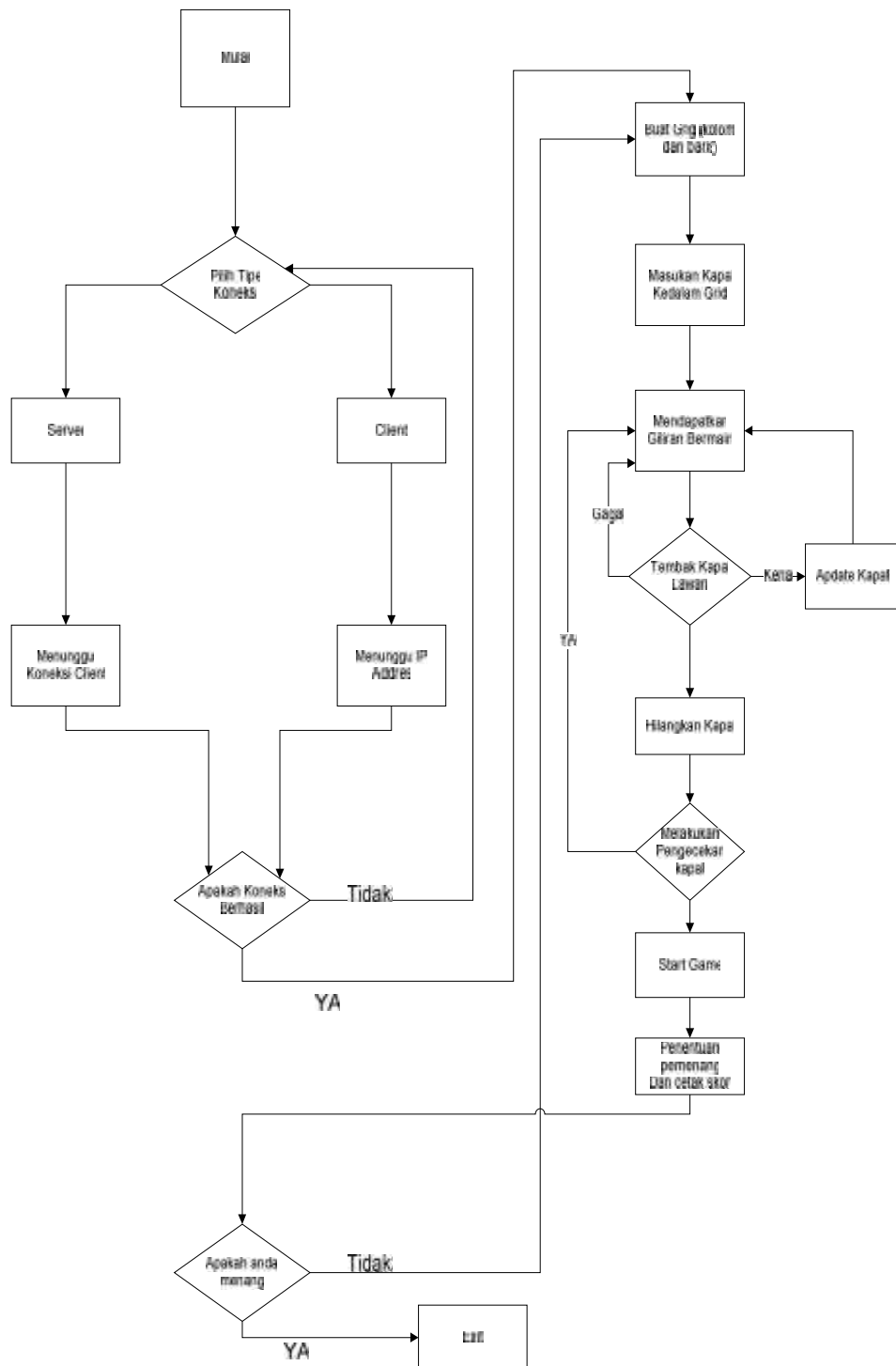


Gambar 4.2 Flowchart permainan Battleship

4.2. Analisa Sistem Baru

Proses analisis kebutuhan mengambil acuan dari *game battleship* dan hasil pengumpulan, pemahaman dan penetapan kebutuhan-kebutuhan (*requirements*) yang ingin didapatkan oleh pengguna. Pada analisis kebutuhan ini diawali dengan identifikasi aktor-aktor yang terlibat dalam *game*, dimana *game* ini kembangkan yang mana *game* yang lama dimaikan dengan satu computer saja. Kemudian sipenulis bagai man mengembangkan *game* ini agar bias dimainkan dengan menggunakan dua computer menggunakan jaringan LAN. Dimana permainan ini menyediakan beberapa buah tempat ynag disusun oleh kotak-kotak sebagai

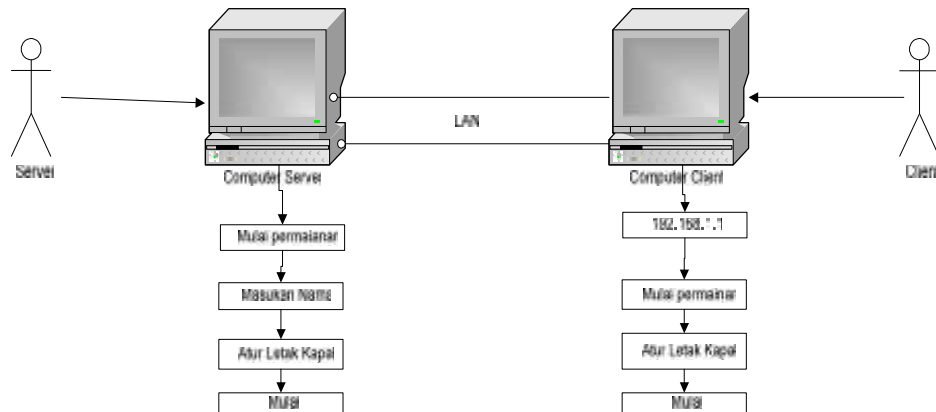
medan perang. Kapal-kapal perang dengan ukuran yang berbeda –beda disusun dalam medan perang masing-masing letak dari kapal k-kapal perang ini tidak terlihat dalam computer pemain lawan. Setiap pemain berusaha untuk menghancurkan kapal-kapal perang pemain lawan. Bagi yang menyisakan maka ialah yang akan menjadi pemenangnya. dan membuat animasi yang lebih bagus dari yang sebelumnya sehingga membuat orang sering memainkan game merasa tidak bosan untuk memainkannya. penjabaran daftar kebutuhan dan kemudian memodelkannya ke dalam suatu diagram *use case*. Analisis kebutuhan ini ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna sesuai *game* yang dibuat. *Flowchart* permainan *battleshipMulti Player* dapat dilihat pada gambar dibawah ini.



Gambar 4.3 *flowchart Game Battleship multi player*

4.2.1. Arsitektur Aplikasi

Berikut adalah gambar arsitektur untuk permainan game *battleship multiplayer*:



Gambar 4.4 Arsitektur aplikasi game *battleship*

Berikut adalah perancangan jumlah kapal yang digunakan untuk game battleship:

- Battleship* : kapal dengan ukuran 4 kotak sebanyak 1 buah.
- Cruiser* : kapal dengan ukuran 3 kotak sebanyak 2 buah.
- Submarine* : kapal dengan ukuran 2 kotak sebanyak 3 buah.
- Destroyer* : kapal dengan ukuran 1 kotak sebanyak 4 buah.

Gambar kapal yang digunakan sebenarnya merupakan penggabungan dari beberapa bagian kapal yang telah dipecahkan sebelumnya. Sebuah gambar kapal dipecahkan menjadi 3 bagian, yaitu bagian depan, tengah dan belakang kapal. Bagian-bagian kapal tersebut digabungkan kembali sesuai dengan ukuran dari kapal dengan perincian sebagai berikut :

- Battleship* merupakan penggabungan dari 1 buah bagian depan kapal, 2 buah bagian tengah kapal dan 1 buah bagian belakang kapal.
- Cruiser* merupakan penggabungan dari 1 buah bagian depan kapal, 1 buah bagian tengah kapal dan 1 buah bagian belakang kapal.

- c. *Submarine* merupakan penggabungan dari 1 buah bagian depan kapal dan 1 buah bagian belakang kapal.

Khusus untuk gambar kapal *Destroyer* yang memiliki panjang 1 kotak saja dirancang dengan menggunakan gambar tersendiri dan bukan merupakan gabungan dari bagian-bagian kapal tersebut.

Gambar-gambar di atas juga dirancang secara vertikal dengan menggunakan metoda yang sama.

4.2.2. Perancangan Animasi dan Suara

a. Animasi

pembuatan animasi pada game *battleship* ini, menggunakan animasi sprite yaitu ada 2 cara yaitu :

- Animasi gerakan
- Animasi bingkai

Gambar kapal yang digerakan secara vertikal dan horizontal , gambar kapal yang terbakar dengan api yang menyala. Gambar kapal terbakar, simbol tembakan tepat sasaran dan tembakan gagal dicari pada sumber-sumber di *internet*, kemudian di-*edit* dan disimpan dengan menggunakan aplikasi *Adobe Photoshop*.



Gambar 4.5 perancangan Animasi

Efek animasi yang digunakan merupakan proses pergantian gambar. Animasi merupakan salah satu bagian grafika komputer yang menyajikan tampilan-tampilan yang sangat atraktif juga merupakan sekumpulan gambar yang ditampilkan secara berurutan dengan cepat untuk mensimulasi gerakan yang hidup. Pemanfaatan animasi dapat ditujukan untuk simulasi, menarik perhatian pemakai komputer pada bagian tertentu dari layar, memvisualisasikan cara kerja suatu alat atau menampilkan keluaran program dengan gambar-gambar yang menarik dibanding dengan sederetan angka, serta tidak ketinggalan untuk program-program permainan.

b. Suara

suara ledakan di cari dari *internet* kemudian diedit dirancang dengan menggunakan aplikasi *Recorder*. Dengan cara memasukkkan suara kedalam rekorder kemudian suara ini nanti akan di cut atau di buat sesuai yang kita inginkan.

4.2.3 Pengaturan Koneksi Setting Client dan Server

Microsoft Winsock berfungsi untuk menghubungkan komputer yang satu dengan yang lainnya dan saling bertukar data dengan menggunakan dua *protokol*/

Algoritmanya adalah sebagai berikut,

{ Tutup winsock untuk menjaga kemungkinan winsock sedang terbuka }

`Winsock.Close`

{ Local Port Default yang dipakai = 1234 }

`Winsock.LocalPort = 1234`

{ Winsock server melakukan aksi listening }

`Winsock.Listen`

Selanjutnya *winsock server* menunggu koneksi dari *winsock client*. *Winsock client* pada komputer lain melakukan koneksi ke *winsock server* dengan meng-input *ip-address* komputer *server* yang dituju. Algoritmanya adalah sebagai berikut,

{ Tutup winsock untuk menjaga kemungkinan winsock sedang terbuka }

```
Winsock.Close
```

```
{Koneksi ke server. Local Port Default yang dipakai = 1234}
```

```
Winsock.Connect <ip-address server>, 1234
```

4.2.4 Perancangan *Interface* untuk Menyusun Kapal

Interface untuk menyusun kapal dirancang dengan ukuran 10 kotak x 10 kotak. 1 kotak diwakili oleh sebuah *picturebox* dengan gambar air, sehingga *interface* ini merupakan penggabungan dari *picturebox* yang diatur posisinya

4.2.4.1 Perancangan Medan Perang

Medan perang dirancang dengan ukuran 10 kotak x 10 kotak. Kotak-kotak ini digambarkan dengan dibatasi oleh garis-garis. Jarak garis-garis tersebut sebanyak 30 *twips* baik secara horizontal maupun vertikal. Sketsa dari gambar medan perang tersebut dapat dilihat pada gambar berikut ini :

	0	30	60	90	120	150	180	210	240	270	300
0											
30											
60											
90											
120											
150											
180											
210											
240											
270											
300											

Gambar 4.6 Sketsa gambar dari medan perang

Kotak-kotak tersebut akan disimpan dalam bentuk *array zone2(x,y)* dengan cara penomoran kotak tersebut adalah sebagai berikut :

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Gambar 4.7 Penomoran kotak pada medan perang

4.2.4.2 Penentuan Posisi Kapal

Kapal yang diletakkan akan ditentukan posisinya dan disimpan ke dalam sebuah variabel *Array* yang memiliki dua buah properti yaitu X untuk menyimpan posisi kolom dari kapal, Y untuk menyimpan posisi baris dari kapal. Proses perhitungan posisi x dan y sebagai berikut :

$$X = \text{Posisi_Kursor_X} \div 30$$

$$Y = \text{Posisi_Kursor_Y} \div 30$$

4.2.4.3 Pengecekan Posisi Tembakan dan Kapal

Jika posisi (X,Y) yang diklik sama dengan salah satu dari posisi kapal yang disimpan dalam *array* tersebut maka berarti kapal tertembak dan posisi bagian kapal yang tertembak tersebut ditandai. Posisi koordinat yang ditembak juga ditandai. Proses perhitungan posisi x dan y tersebut sama dengan proses perhitungan pada penentuan posisi kapal di atas.

4.3 Cara bermain game *battleship Multiplayer*

Kapal-kapal perang dengan ukuran yang berbeda-beda disusun dalam medan perang masing-masing. Letak dari kapal-kapal perang ini tidak terlihat dalam komputer pemain lawan. Setiap pemain berusaha untuk menghancurkan kapal-kapal perang pemain lawan dengan cara meng-klik kotak-kotak yang dianggap sebagai letak dari kapal-kapal perang lawan. Setiap pemain hanya boleh

menebak satu kali saja setiap gilirannya. Pemain yang masih menyisakan kapal perang yang memenangkan permainan (*game*). Berikut langkah-langkah permainan game battleship.

1. *Login*

Pada tahap login akan muncul tampilan menu utama

2. Pilih tipe koneksi

Apabila *user* memilih koneksi sebagai server, maka harus menunggu koneksi dari pemain client. Apabila user memilih koneksi sebagai client, maka harus memasukkan ip-address computer server yang dituju.

3. Apabila telah terhubung maka muncul tampilan *from* untuk meng-*input* nama pemain.

4. Setelah input nama, maka akan muncul tampilan pengaturan letak kapal
Peletakan kapal bisa diatur secara horizontal atau vertikal.

5. Mulai permainan

4.3.1 Penentuan Pemenang

Jika jumlah bagian kapal yang ditandai sama dengan 20 maka berarti semua kapal telah tertembak, maka pemain lawan dinyatakan menang.

4.3.2 Perhitungan Score

Jika sebuah kapal telah ditenggelamkan maka *score* pemain akan ditambahkan. Besar *score* yang didapatkan dihitung berdasarkan ukuran kapal. Ukuran 1 kotak bernilai 10 point. Jadi, menenggelamkan sebuah kapal berukuran 3 kotak, maka pemain akan mendapatkan nilai 30 *point*.

4.4 Perancangan Sistem

Unified Modeling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk menentukan, visualisasi, merancang dan mendokumentasikan artifact dari sistem software, dan untuk memodelkan system. UML menyediakan beberapa notasi dan artifact standar yang bisa digunakan

sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain system pelayanan *customer* online.

Artifact didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. UML (Astah) adalah model perancangan sistem berorientasi objek, model perancangan sistem ini membahas model perancangan sebagai berikut :

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*

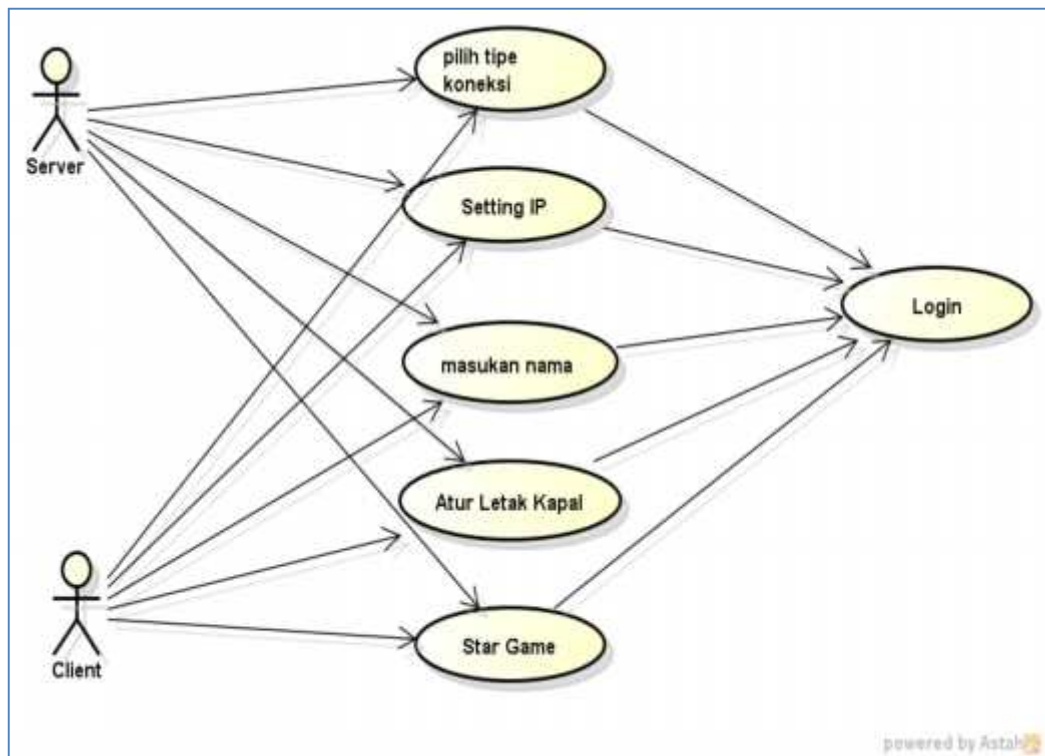
Berikut ini adalah rancangan system yang akan dibangun dan di visualisasikan dalam beberapa diagram :

4.4.1 Perancangan *Use Case Diagram*

Use Case menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case mempresentasikan sebuah interaksi antara actor dengan sistem.

Pemaparan dari use case dibawah adalah sebagai berikut:

- 1) Pilih Tipe Koneksi, user memilih sebagai Server dan Client
- 2) Setting IP Game, user memasukan ip. lawan
- 3) Masukan nama , user dapat memasukan nama
- 4) Atur letak kapal, user dapat mengatur letak kapal
- 5) Start game, user memulai permainan



Gambar 4.8 Use Case BattleShip login

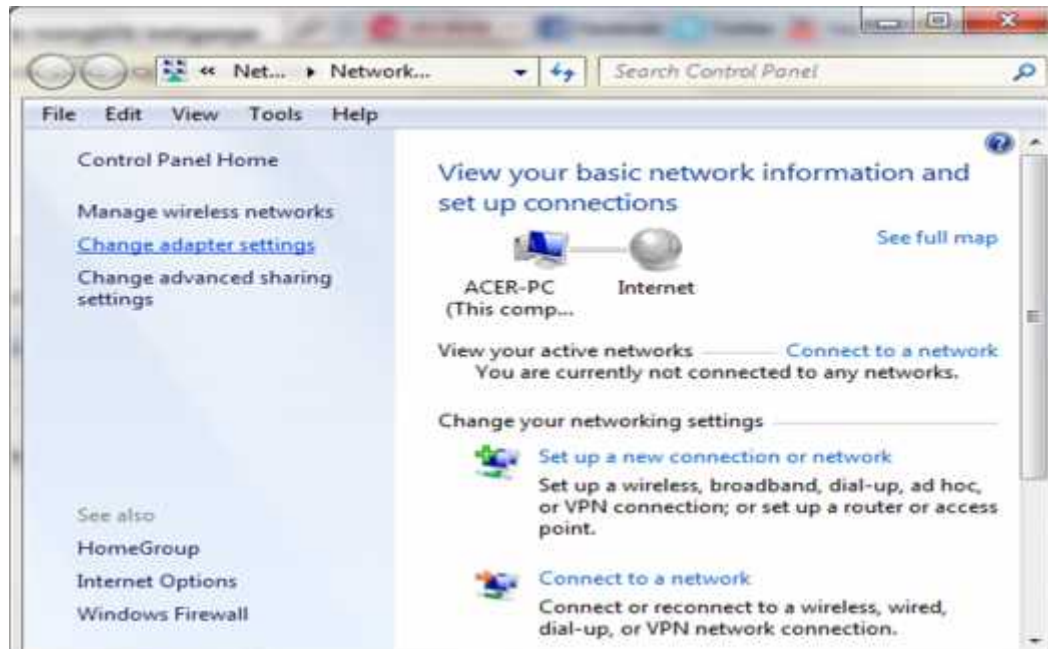
Berikut adalah langkah-langkah setting client dan Server:

1. langkah pertama yang harus dilakukan klik open Network and sharing center.



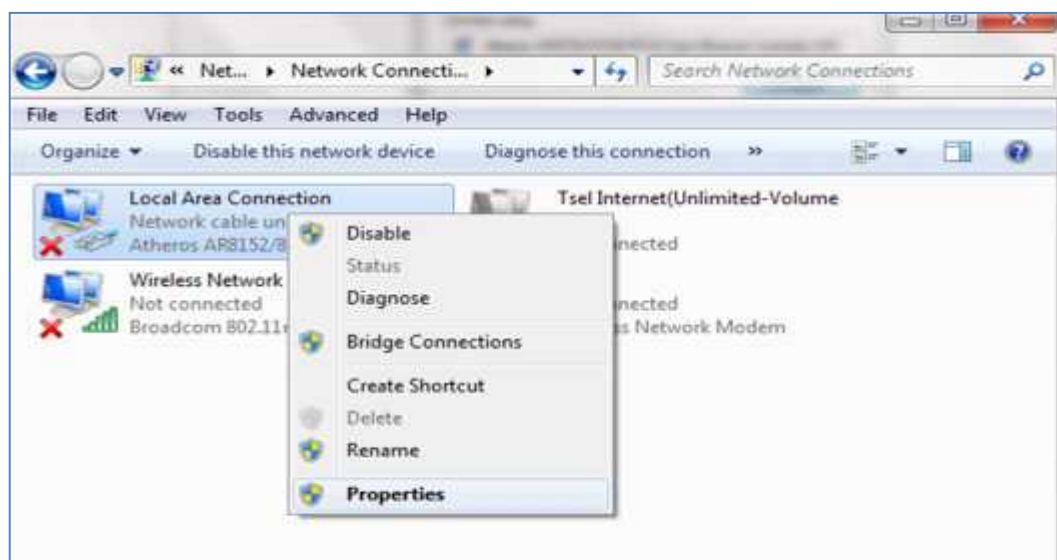
Gambar 4.9 Open Network and Sharing Center

2. Selanjutnya akan muncul jendela *Networks and Sharing Center*. Pada sisi sebelah kiri jendela ini, klik tulisan *Change Adapter Setting*



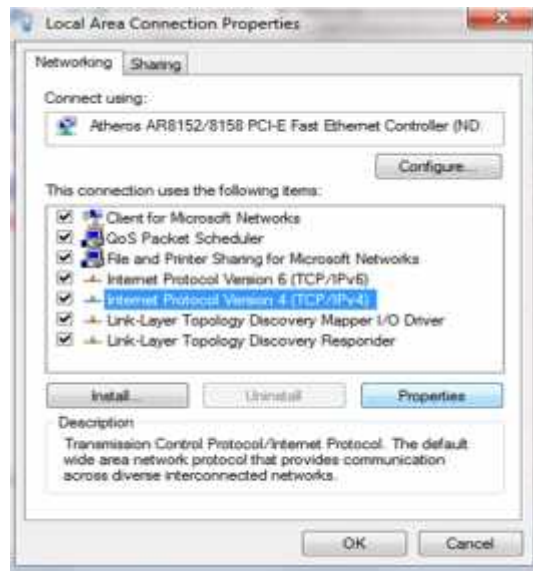
Gambar 4.10 Change Adapter Setting

3. Klik kanan pada Network Card Anda dan pilih properties



Gambar 4.11 Local Area Connection

4. Pada jendela Local Area Connection Properties, pilih Internet Protocol (TCP/IP) pada Windows XP atau Internet Protocol Version 4 (TCP/IPv4) pada Windows 7 dan Vista. Kemudian klik tombol properties.



Gambar 4.12 Local Area Connection Propertis

5. Pada jendela properties yang muncul, pilih opsi Use the followinf IP Address dan isikan dengan 192.168.0.1 pada IP Adrees, 255.255.255.0 pada Subnetmask. Input yang tersisa bisa Anda kosongkan.



Gambar 4.13 Masukan IP Adrees

6. untuk komputer ke 2 nya, anda settingkan seperti gambar dibawah ini
Kemudian Pilih OK.

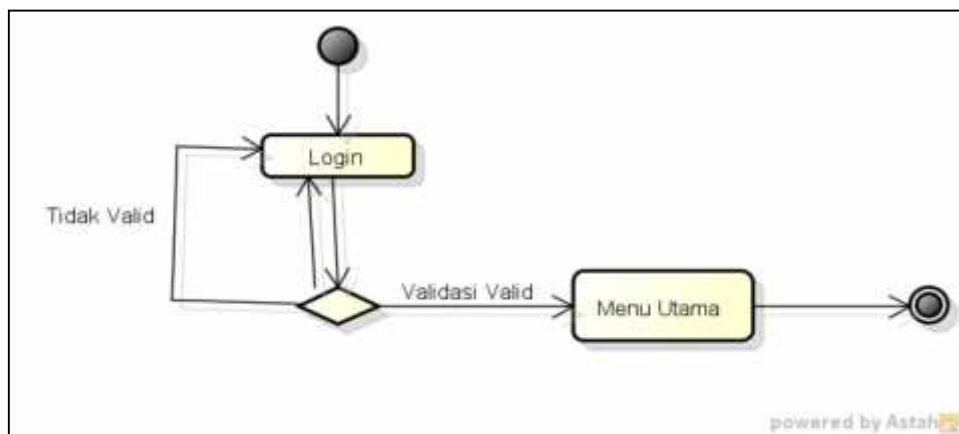


Gambar 4.14 Masukan IP Adres

4.4.2 Activity Diagram

4.4.2.1 Activity Diagram untuk Login

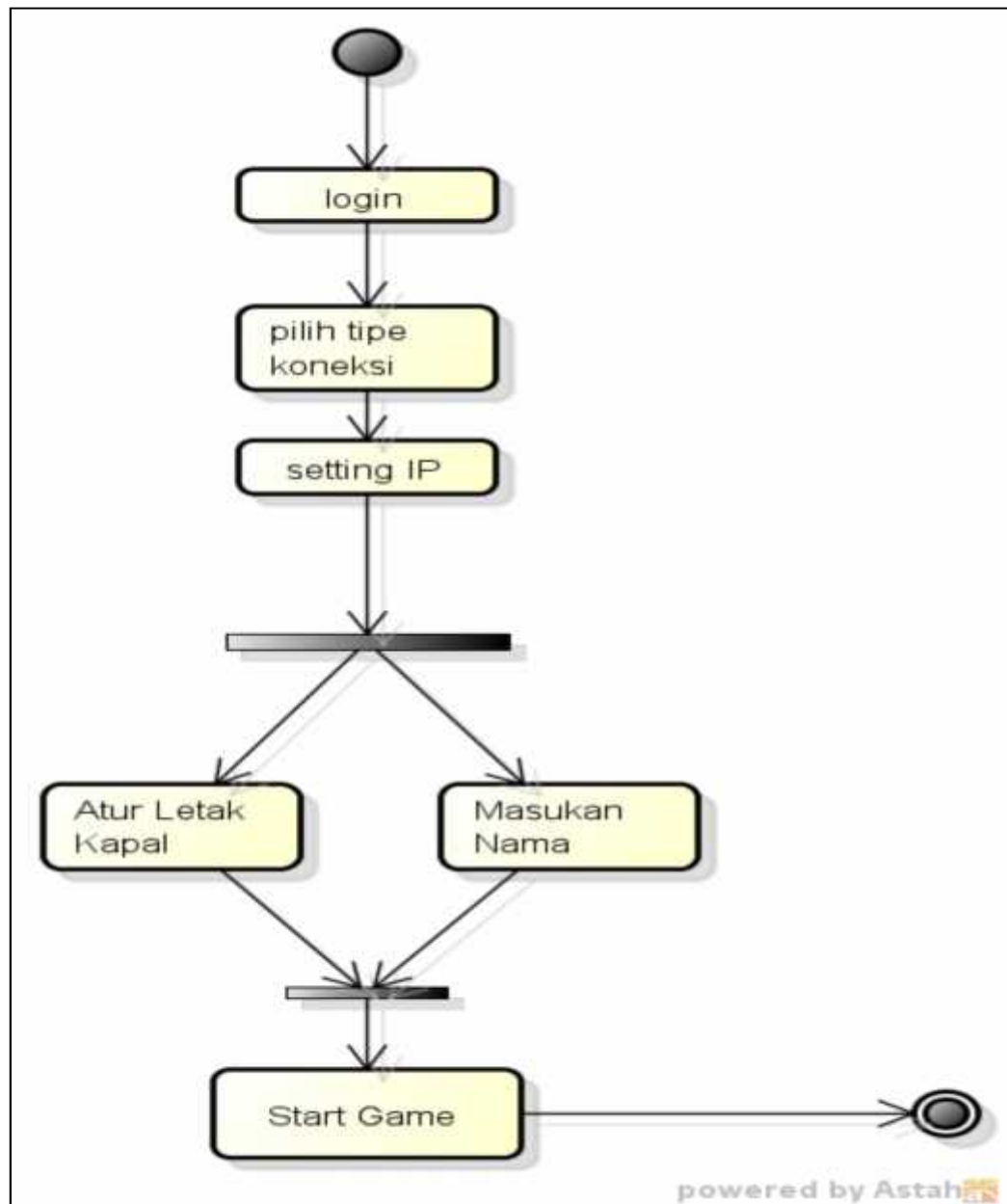
Activity Diagram login digunakan untuk menggambarkan jalur kerja sistem pada saat pertama kali *user* menjalankan sistem pelayanan *customer* maka terlebih dahulu harus melakukan *login* dengan mengisi *username* dan *password* masing-masing. Jika *username* dan *password* yang diisikan benar, maka sistem akan menampilkan menu utama sesuai dengan hak akses *user*.



Gambar 4.15 Activity Diagram untuk Login

A. Server melakukan permainan

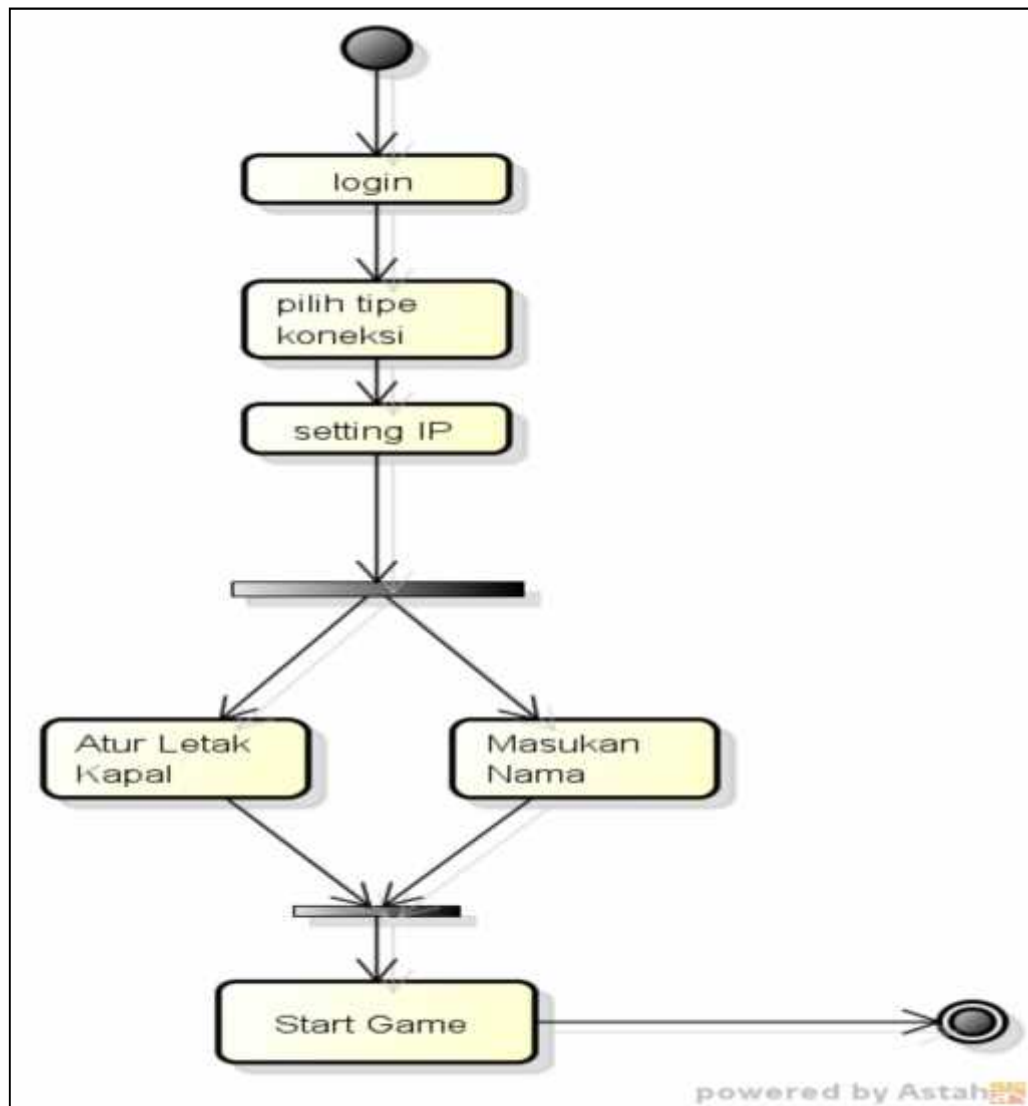
Diagram activity Server menggambarkan jalur kerja system saat melakukan permainan game



Gambar 4.16 Activity Diagram melakukan Permainan

B. Client melakukan permainan

Diagram activity client menggambarkan jalur kerja system saat melakukan permainan game.



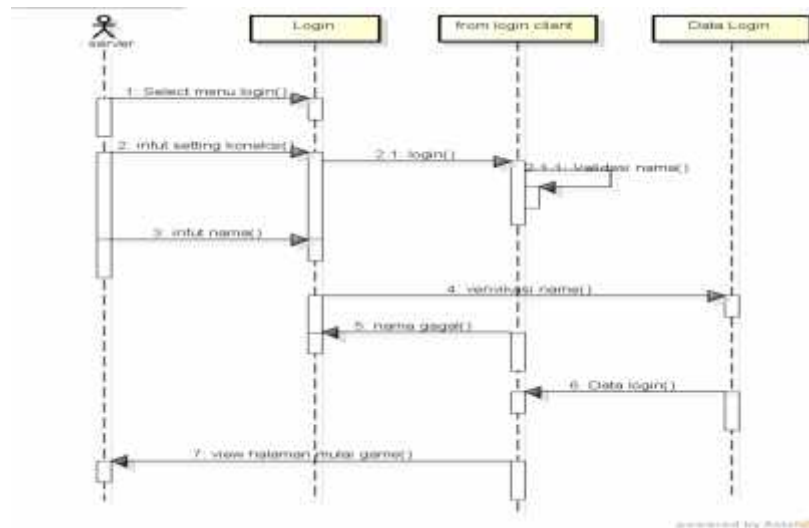
Gambar 4.17 Activity Diagram melakukan Permainan

4.4.3 Perancangan Sequence Diagram

Sequence diagram digunakan untuk memberikan gambaran detail dari setiap use case diagram yang telah dibuat sebelum nya. Setiap *objects* yang terlibat dlam sebuah use case digambarkan dengan garis putus-putus vertical,

kemudian message yang dikirim oleh objek digambarkan dengan garis horizontal secara kronologis dari atas kebawah

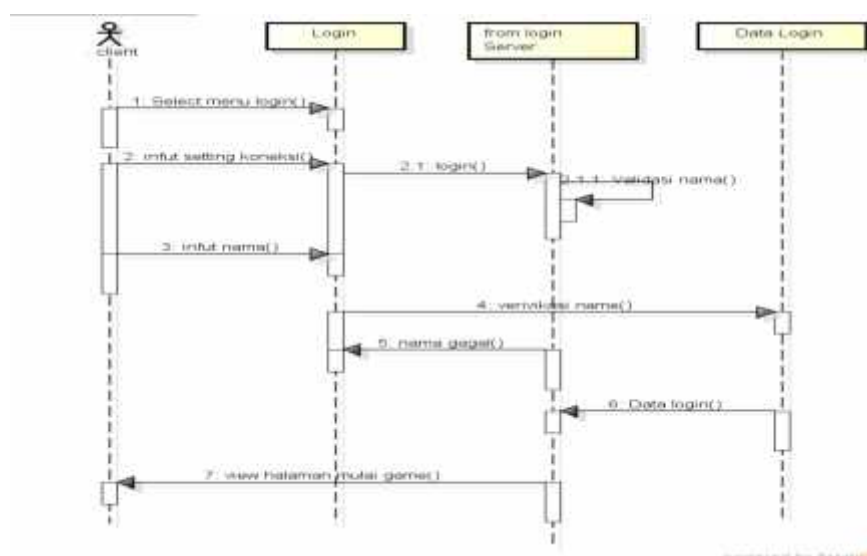
a. Sequence Diagram *Login Server*



Gambar 4.18 *Sequence Diagram login Server*

Seorang user ketika memulai game akan masuk kemenu utama game Battleship. Pada menu utama terdapat beberapa menu yang akan diinputkan oleh user untuk memainkan game battleship.

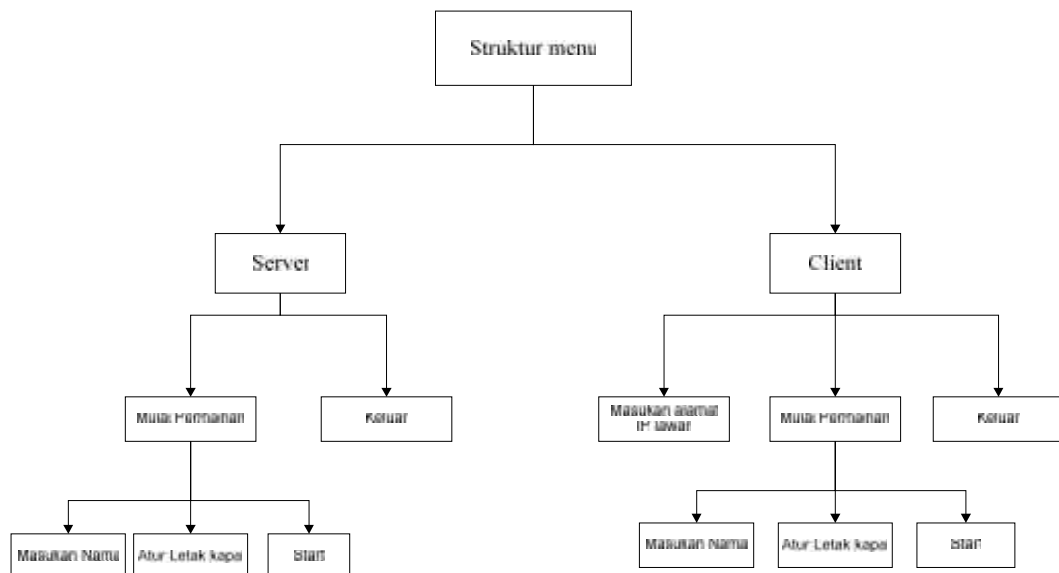
b. Sequence Diagram *Client*



Gambar 4.18 *Sequence Diagram login Client*

4.5 Perancangan Struktur Menu

Pada tahap ini dilakukan perancangan menu untuk game battleship multi player menu untuk halaman utama game ini dapat dilihat pada gambar berikut:



Gambar: 4.19 Struktur menu game *battleship*

4.5.1 Perancangan *Splash Screen*

Berikut ini adalah tampilan *splash screen battleship multiplayer*:

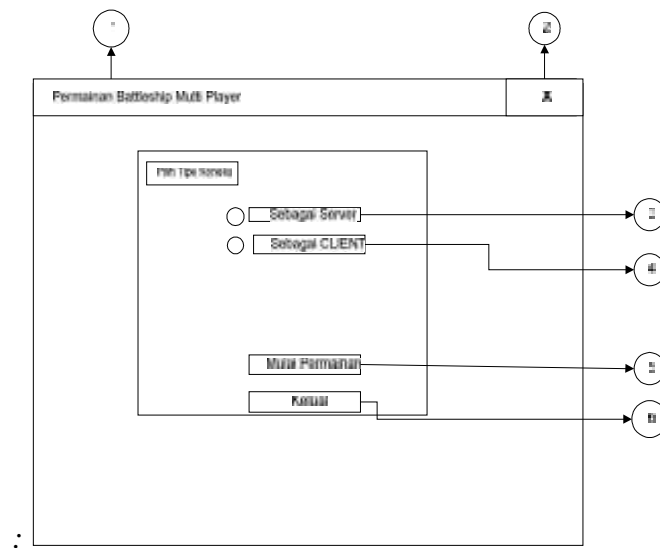
Background	Judul aplikasi
Animasi	Copyright @

Gambar 4.20 Tampilan *Splash Screen Battleship Multiplayer*

4.5.2 Perancangan Pengaturan Koneksi.

Pada tampilan menu koneksi terdapat menu server dan client untuk memulai permainan game apakah bertindak sebagai server atau client. tampilan menu server atau client dapat dilihat pada gambar berikut:

4.5.2.1 Tampilan perancangan menu Server



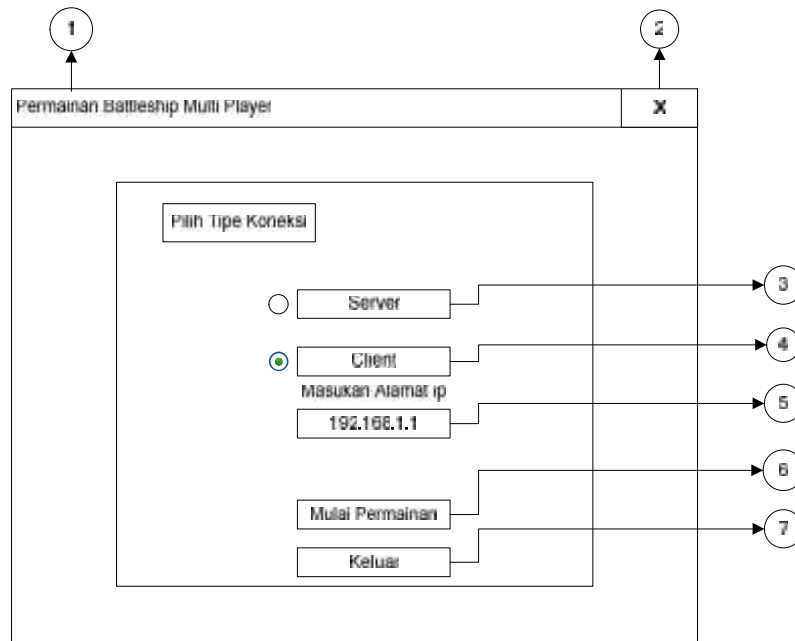
Gambar 4.21 Tampilan Pengaturan Koneksi *Battleship Multiplayer*

Keterangan :

1. *title bar*.
2. tombol 'x', berfungsi untuk menutup perangkat lunak.
3. *option button* 'Sebagai Server' untuk bermain pada jaringan dengan koneksi sebagai *server*.
4. *option button* 'Sebagai Client' untuk bermain pada jaringan dengan koneksi sebagai *client*.
5. tombol 'Mulai Permainan', berfungsi untuk menampilkan tampilan Tempat Penyusunan Kapal.
6. tombol 'Keluar', berfungsi untuk keluar dari perangkat lunak.

4.5.2.2 Tampilan perancangan Client

Berikut adalah tampilan menu perancangan menu client pada game battleship multi player:



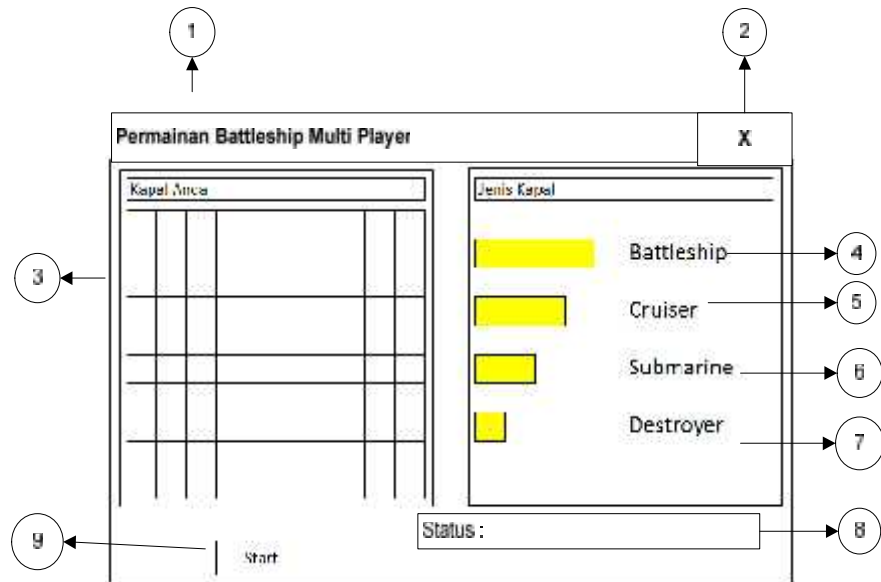
Gambar 4.22 Tampilan Pengaturan Client *Battleship Multiplayer*

Keterangan :

1. *title bar*.
2. tombol 'x', berfungsi untuk menutup perangkat lunak.
3. *option button* 'Sebagai Server' untuk bermain pada jaringan dengan koneksi sebagai *server*.
4. *option button* 'Sebagai Client' untuk bermain pada jaringan dengan koneksi sebagai *client*.
5. Tombol Alamat IP' berfungsi untuk memasukan alamat ip nya.
6. tombol 'Mulai Permainan', berfungsi untuk menampilkan tampilan Tempat Penyusunan Kapal.
7. tombol 'Keluar', berfungsi untuk keluar dari perangkat lunak.

4.5.3 Perancangan Tempat Penyusunan Kapal

Berikut ini adalah tampilan tempat penyusunan kapal permainan *battleship multiplayer*.



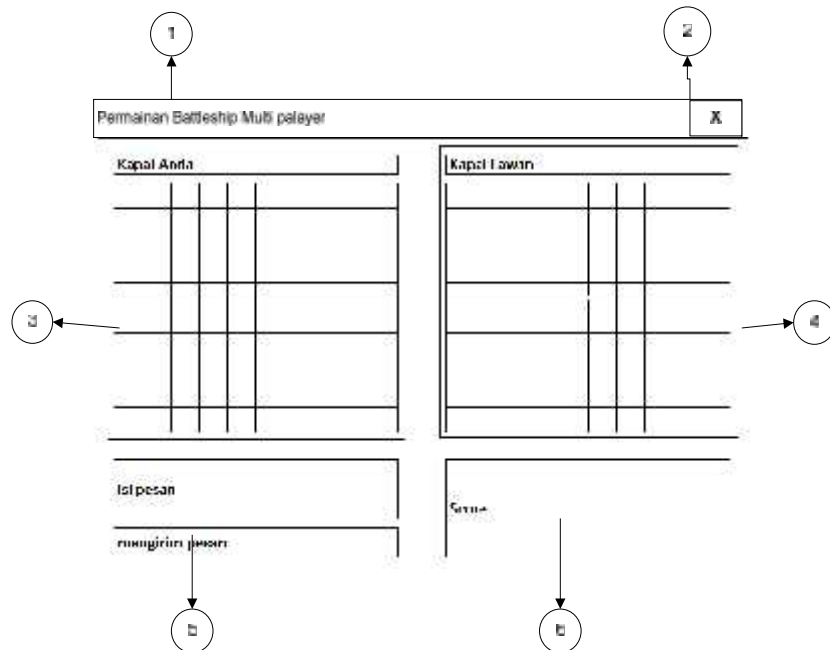
Gambar 4.23 Tampilan Tempat Penyusunan Kapal Battleshi Multiplayer

Keterangan :

1. *title bar*.
2. tombol 'x', berfungsi untuk menutup perangkat lunak.
3. tempat penyusunan kapal.
4. *option button* 'Battleship' untuk memilih kapal *Battleship*.
5. *option button* 'Cruiser' untuk memilih kapal *Cruiser*.
6. *option button* 'Submarine' untuk memilih kapal *Submarine*.
7. *option button* 'Destroyer' untuk memilih kapal *Destroyer*.
8. tombol 'Mulai' untuk memulai permainan.
9. status giliran dari pemain yaitu apakah sudah siap atau belum.

4.5.4 Perancangan medan perang Medan Perang

Berikut ini adalah tampilan medan perang permainan *battleship multiplayer*.



Gambar 4.24 Tampilan Medan Perang *Battleship Multiplayer*

Keterangan :

1. *title bar*.
2. tombol 'x', berfungsi untuk menutup perangkat lunak.
3. daerah tampilan posisi kapal pemain.
4. tempat pengisian pesan yang ingin dikirimkan kepada pemain lawan.

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Sistem

Implementasi siste dalam perangkat lunak pembelajaran ini mencakup spesifikasi kebutuhan perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*).

5.1.1 Lingkungan Implementasi

Pada prinsipnya setiap desain sistem yang telah dirancang memerlukan sarana pendukung yaitu berupa peralatan-peralatan yang sangat berperan dalam menunjang penerapan sistem yang didesain terhadap pengolahan data. Komponen-komponen yang dibutuhkan antara lain *hardware*, yaitu kebutuhan perangkat keras komputer dalam pengolahan data, kemudian *software* yaitu untuk kebutuhan akan perangkat lunak berupa sistem untuk mengoperasikan sistem yang telah didesain.

Lingkungan operasional merupakan lingkungan tempat sistem ini dapat digunakan oleh satu orang pengguna. Berikut merupakan spesifikasi minimum kebutuhan dari lingkungan operasional yaitu:

5.1.2 Kebutuhan Perangkat Keras (*Hardware*)

1. *Prosesor Intel Pentium IV 2.4 GHz.*
2. *Memory 128 MB.*
3. *Harddisk 40 GB, dengan freespace 200 MB.*
4. *VGA card 32 MB.*
5. *Monitor dengan resolusi 800 X 600 pixel.*
6. *Keyboard dan Mouse*
7. *Network Card – 100 MBps dan adanya jaringan LAN.*

5.1.3 Kebutuhan Perangkat Lunak (*Software*)

Adapun perangkat lunak (*software*) yang direkomendasikan untuk menjalankan aplikasi ini adalah:

- a. Visual Basic 6.0 untuk pembuatan Game
- b. Adobe Photoshop CS4 membuat Karakter
- c. Microsoft Windows 7 Ultimate

5.2 Tampilan Menu Utama

Klik pada file '*Permainan Battle-Ship (Multiplayer).exe*' untuk menjalankan perangkat lunak. Setelah itu akan muncul tampilan *Splash Screen*. Klik pada *form* atau tekan '*Enter*' untuk masuk ke dalam *program*.



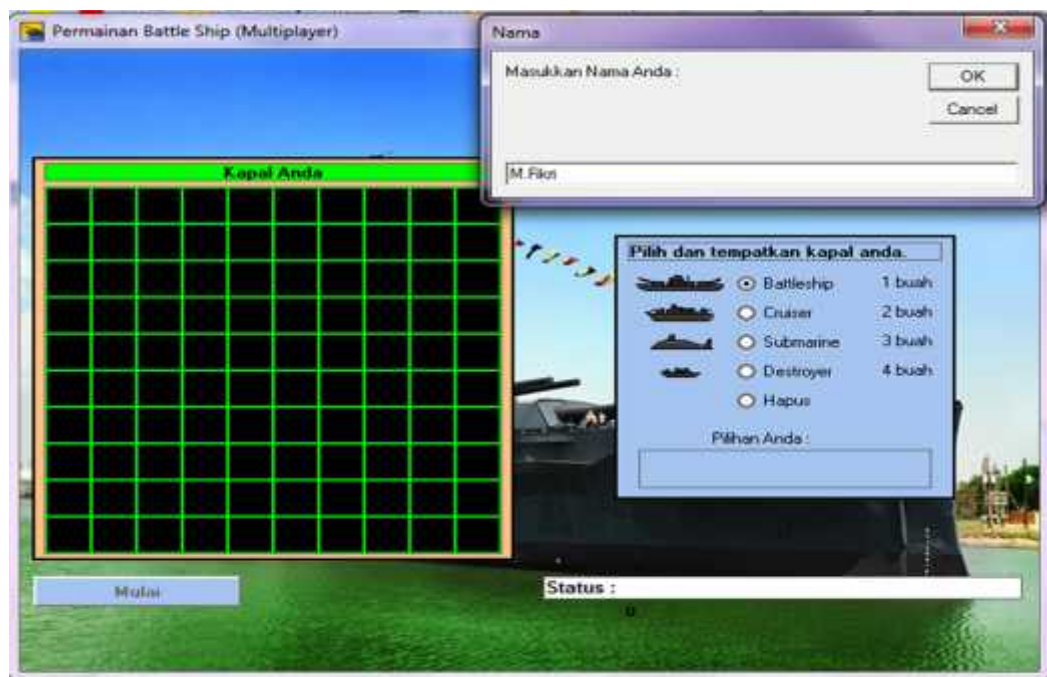
Gambar 5.1 Tampilan *Form Splash Screen*

Setelah itu, akan muncul tampilan '*Setting Koneksi*'. Apabila anda memilih koneksi sebagai *server*, maka anda harus menunggu koneksi dari pemain *client*. Apabila anda memilih koneksi sebagai *client*, maka anda harus memasukkan *ip-address* komputer *server* yang dituju.



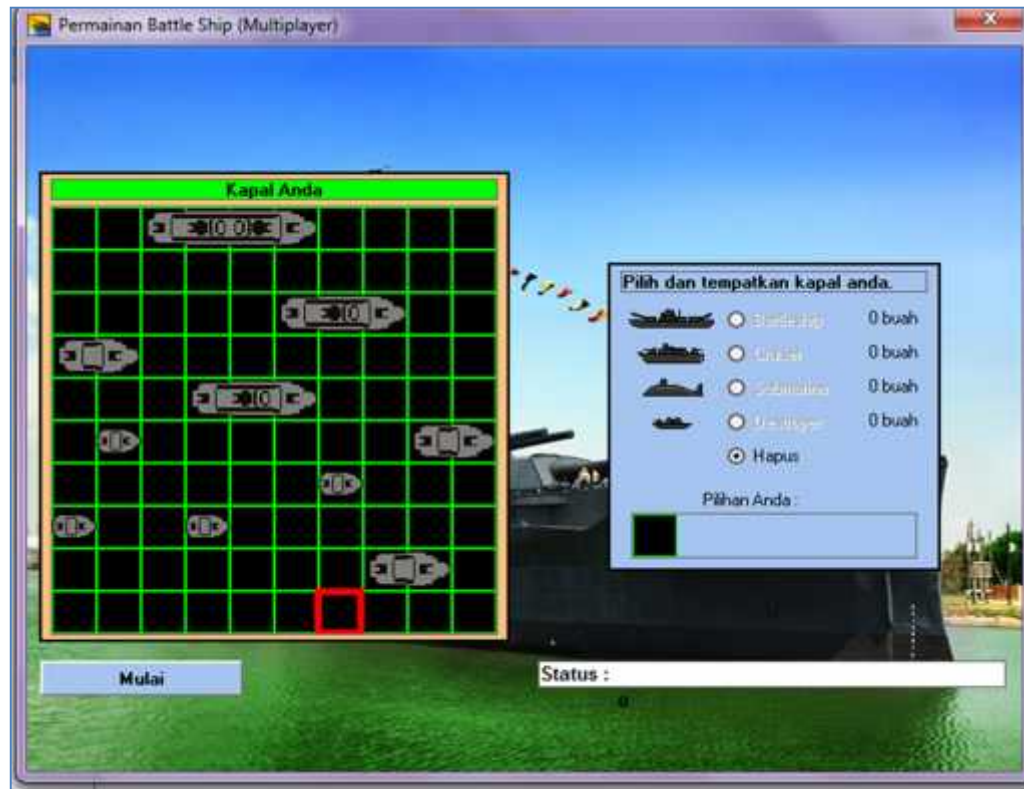
Gambar 5.2 Tampilan *Setting* Koneksi

Apabila koneksi *winsock* telah terhubung, maka akan muncul tampilan *input form* untuk meng-*input* nama pemain.



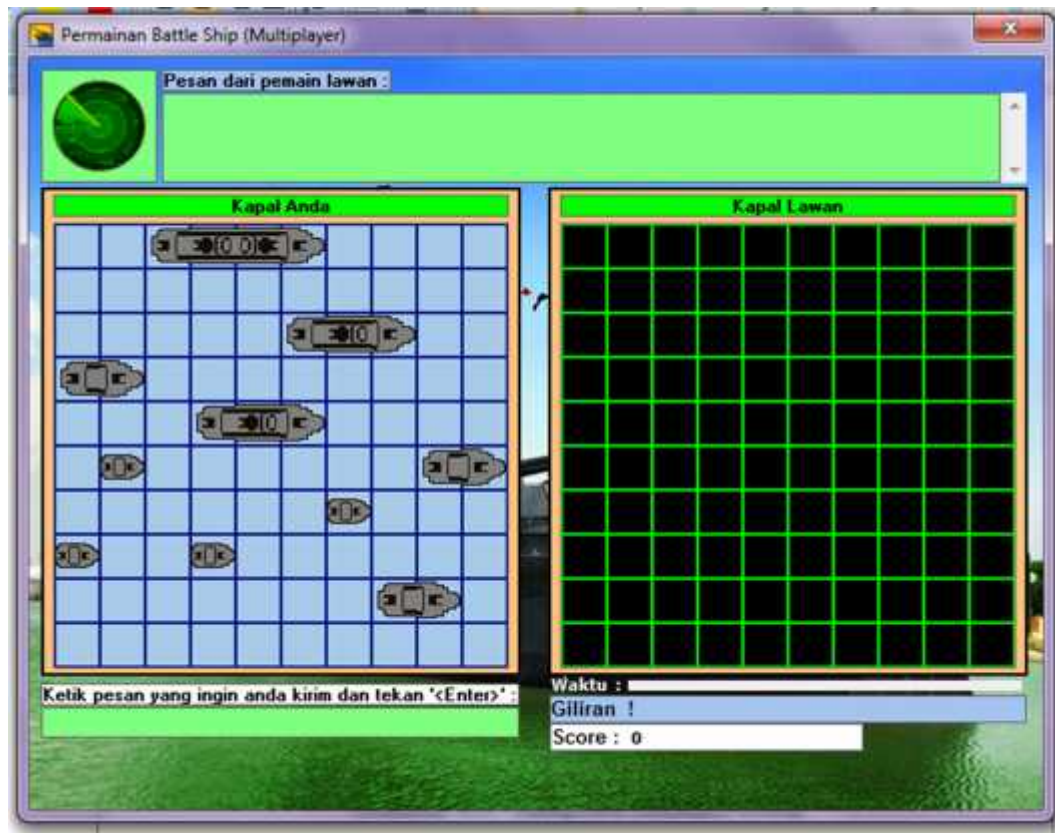
Gambar 5.3 Tampilan *Input Form* Nama

Setelah *input* nama pemain, maka akan muncul tampilan ‘Pengaturan Letak Kapal’. Arah letak kapal dapat diatur secara horizontal atau vertikal. Klik kanan *mouse* pada daerah penempatan kapal untuk mengubah arah horizontal menjadi vertikal, dan sebaliknya.



Gambar 5.4 Tampilan Pengaturan Letak Kapal

Permainan akan dimulai setelah kedua pemain klik tombol ‘Mulai’. Selanjutnya, muncul tampilan medan perang.



Gambar 5.5 Tampilan Medan Perang

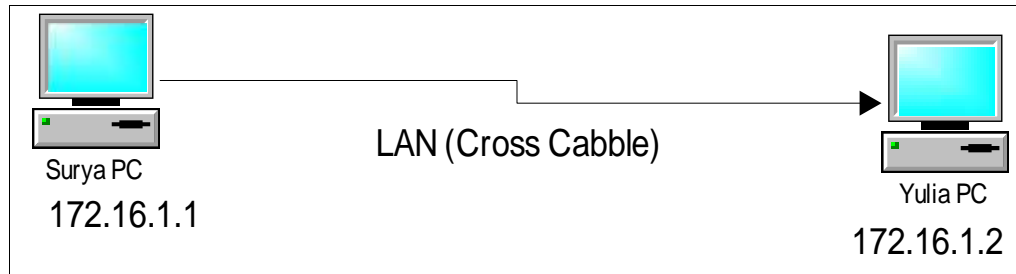
Daerah kotak sebelah kiri merupakan letak kapal anda, sedangkan daerah kotak sebelah kanan merupakan letak kapal lawan. Pada giliran anda, klik kiri pada kotak sebelah kanan untuk menembak kapal lawan.

Apabila tembakan gagal (tidak mengenai kapal lawan), maka akan muncul animasi gelombang air pada daerah kotak sebelah kanan.

5.3 Pengujian Sistem

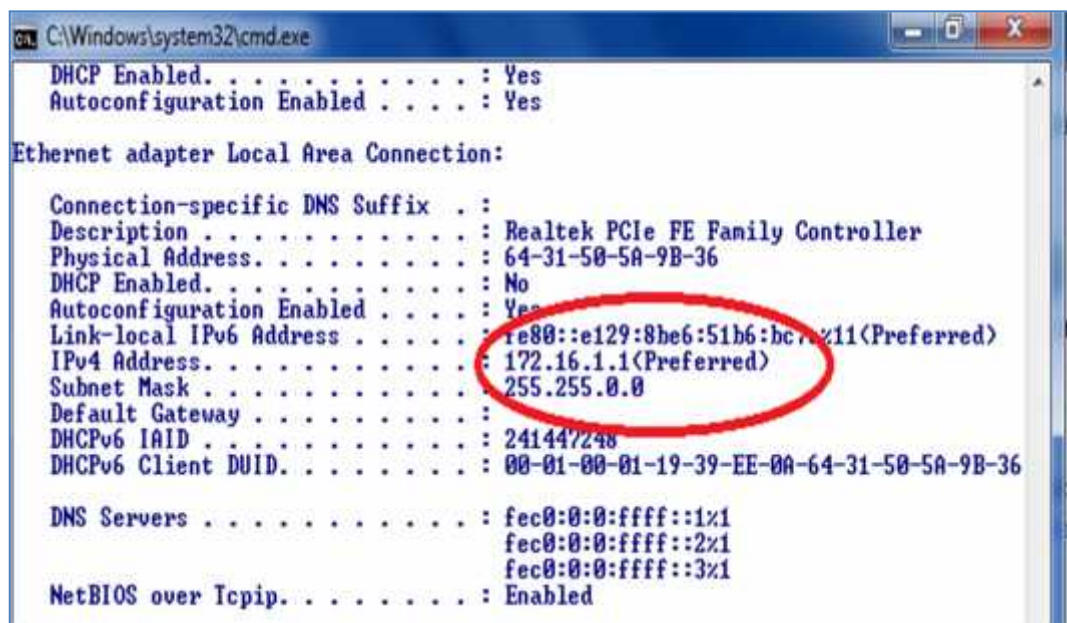
Pemograman merupakan kegiatan penulisan kode program yang akan dieksekusi oleh komputer berdasarkan hasil dari analisa dan perancangan sistem. Sebelum program diimplementasikan, maka program tersebut harus bebas dari kesalahan pengujian program dilakuakn untuk menemukan kesalahan-kesalahan yang mungkin terjadi.

5.3.1 Skema pengujian 1 : koneksi LAN (*Local Area Network*)



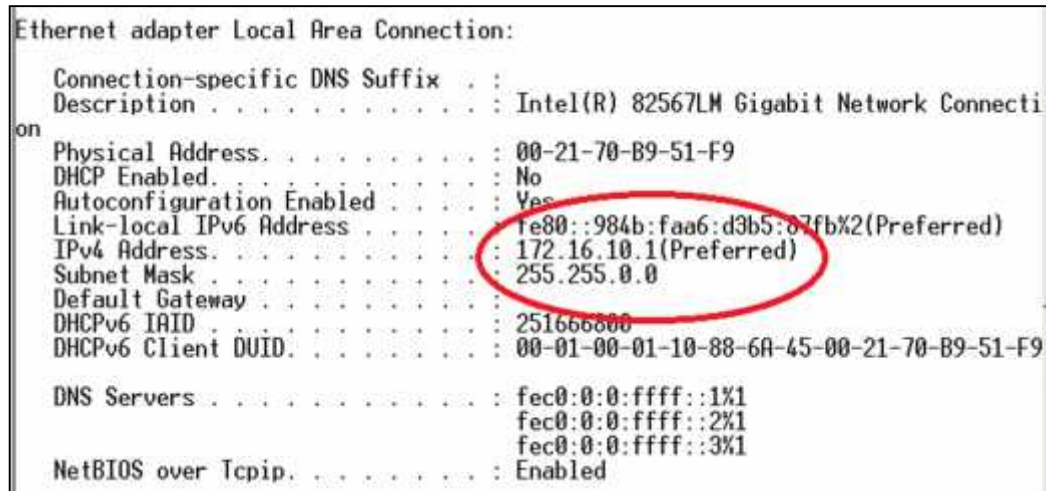
Gambar 5.6 pengujian menggunakan jaringan LAN

Setting IP untuk Yulyahr PC Menggunakan IP address private : 172.16.1.1
Subnetmask 255.255.0.0 (default)



Gambar 5.7 Setting IP address untuk komputer Yulyahr-PC untuk koneksi melalui LAN

Setting IP untuk Surya-PC Menggunakan IP address private : 172.16.10.1
Subnetmask 255.255.0.0 (default)

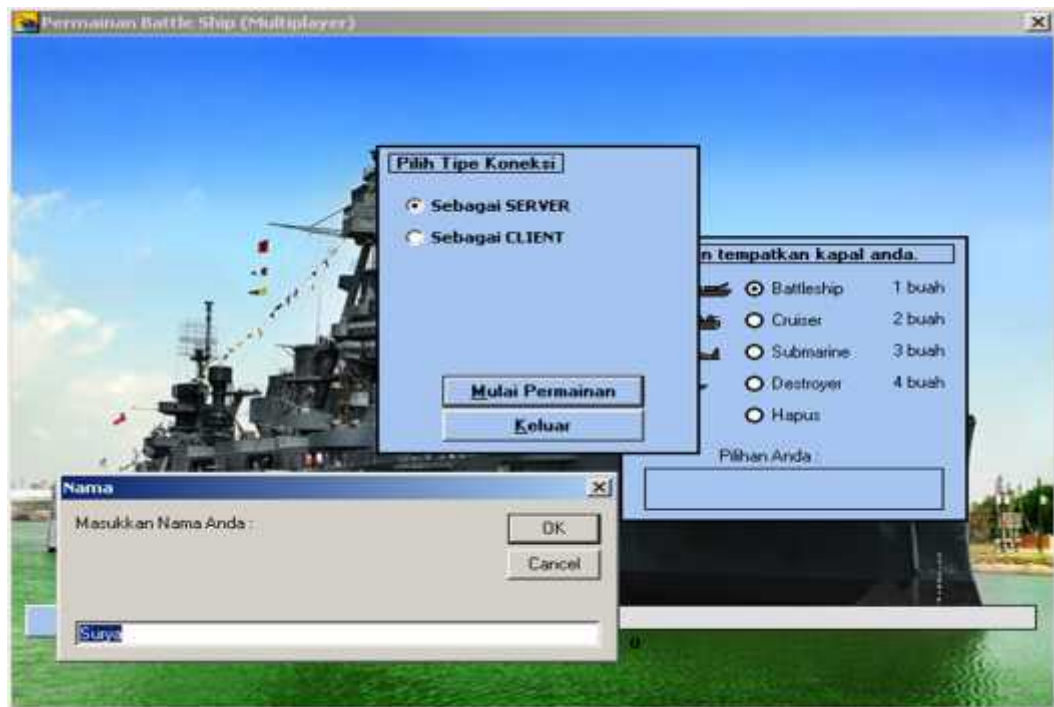


Gambar 5.8 Setting IP address untuk komputer Surya-PC untuk koneksi melalui LAN

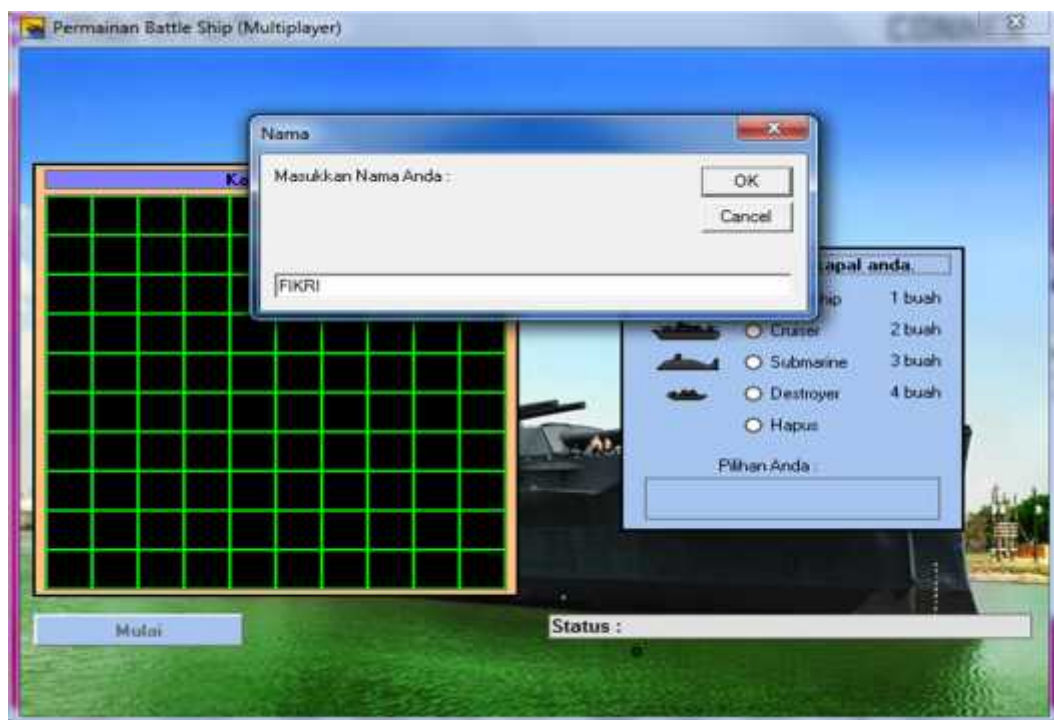
Setelah terhubung dengan jaringan LAN (dalam pengujian ini dilakukan koneksi point-to-point menggunakan kabel UTP cross), dilakukan percobaan untuk menghubungkan kedua komputer pada game Battleship, bergantian salah satu sebagai server, yang lain sebagai klien, dan sebaliknya.

A. Komputer Surya-PC sebagai Server, Yulyahr-PC sebagai Klien

Untuk pertama sekali, dicoba komputer Surya-PC sebagai server, dan Yulyahr-PC sebagai klien. Proses koneksi berhasil dengan lancar karena koneksi point-to-point yang memiliki reliabilitas sangat tinggi, dan tidak ada firewall yang membatasi jenis trafik maupun bandwidth. Hal ini ditunjukkan saya pada Gambar 5.9 dan 5.10 berikut ini.



Gambar 5.9 Komputer Surya-PC bertindak sebagai server, dan berhasil menjalin koneksi ke komputer klien melalui LAN



Gambar 5.10 komputer Yulhar-PC sebagai client, berhasil menghubungi server melalui LAN

Untuk membuktikan koneksi yang berhasil terbentuk, dilakukan monitoring status koneksi TCP melalui perintah netstat, sebagai mana gambar xx dan xx berikut ini. Dari kedua gambar tersebut, terlihat bahwa antara PC server (IP 172.16.10.1) membuka koneksi TCP pada port 1234, yaitu port yang diset pada saat pembuatan koding game Battleship.

```
C:\Users\YULYA HR>netstat
Active Connections

```

Proto	Local Address	Foreign Address	State
TCP	172.16.1.1:49231	Surya-PC:1234	ESTABLISHED

```
C:\Users\YULYA HR>
```

Gambar 5.11 Komputer Yulyahr-PC sebagai klien berhasil menghubungi komputer server game pada port 1234

Koneksi TCP dari klien yang ingin bermain game Battleship selanjutnya akan masuk melalui nomor port 1234 tersebut. Hal ini ditunjukkan oleh gambar xx pada komputer klien yang menghubungi server melalui port 4931.

```
C:\Users\Surya>netstat
Active Connections

```

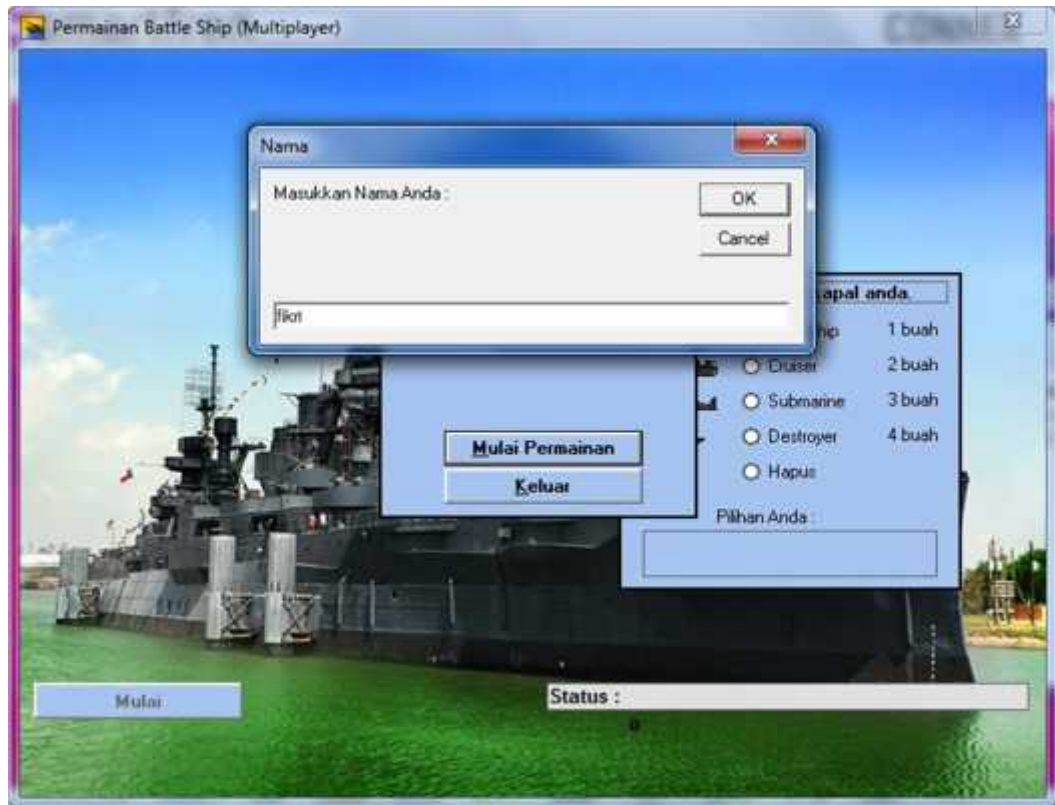
Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:3979	Surya-PC:49158	ESTABLISHED
TCP	127.0.0.1:3979	Surya-PC:49159	ESTABLISHED
TCP	127.0.0.1:5357	Surya-PC:53225	TIME_WAIT
TCP	127.0.0.1:49158	Surya-PC:3979	ESTABLISHED
TCP	127.0.0.1:49159	Surya-PC:3979	ESTABLISHED
TCP	127.0.0.1:52126	Surya-PC:52127	ESTABLISHED
TCP	127.0.0.1:52127	Surya-PC:52126	ESTABLISHED
TCP	127.0.0.1:53225	Surya-PC:msd	TIME_WAIT
TCP	172.16.10.1:1234	YULYAH-PC:49231	ESTABLISHED

```
C:\Users\Surya>
C:\Users\Surya>
```

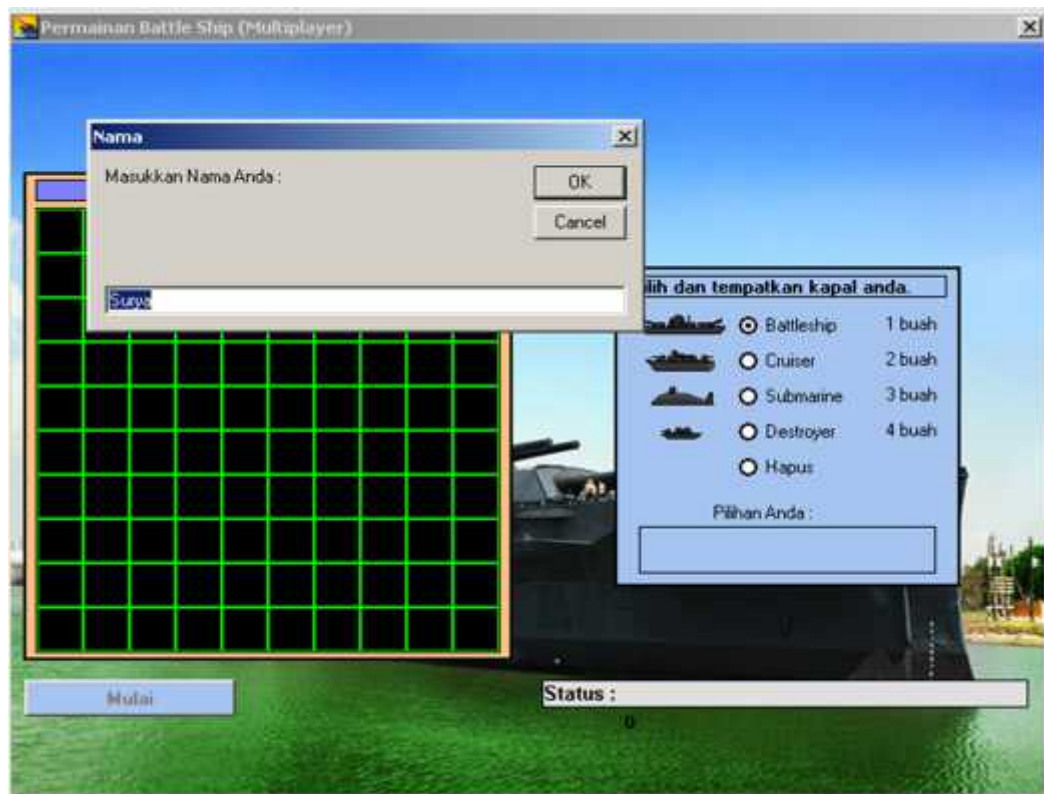
Gambar 5.12 Status koneksi TCP dari komputer Surya-PC sebagai server, berhasil dihubungi oleh klien melalui LAN

B. Komputer Surya-PC sebagai Klien, dan Yulyahr-PC sebagai server

Pengujian kedua, Sebagai server adalah Yulyahr PC. Hubungan koneksi game battle ship pun dapat dilakukan dengan lancar, sebagaimana ditunjukkan pada gambar 5.13 dan 5.14 berikut ini.



Gambar 5.13 Komputer Yulhar-PC sebagai server sudah berhasil dihubungi oleh klien melalui koneksi LAN

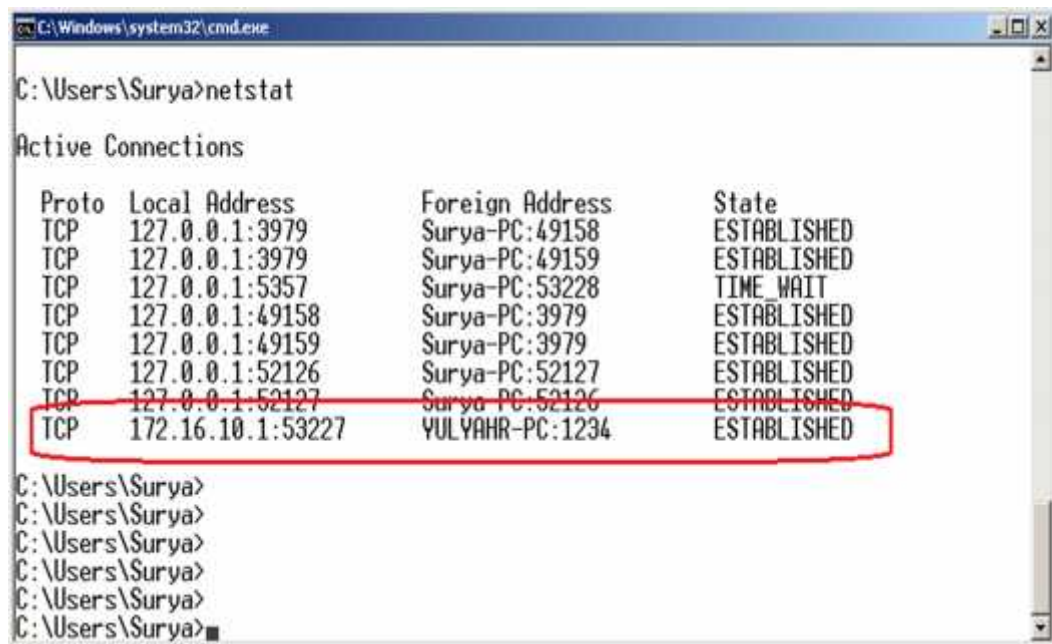


Gambar 5.14 komputer Surya-PC sebagai klien yang terhubung LAN sudah berhasil menjalin koneksi ke komputer server

Pada saat game telah dimainkan, user telah memasukkan nama pemain dan mulai memilih lokasi kapal, dilakukan monitoring koneksi TCP untuk memperlihatkan status koneksi yang terjadi antara komputer klien (Surya-PC) dan server (Yulyahr-PC), seperti ditunjukkan oleh gambar 5.15 dan 5.16 berikut ini.



Gambar 5.15 Status TCP pada Komputer server (Yulyahr-PC) pada saat koneksi game melalui LAN

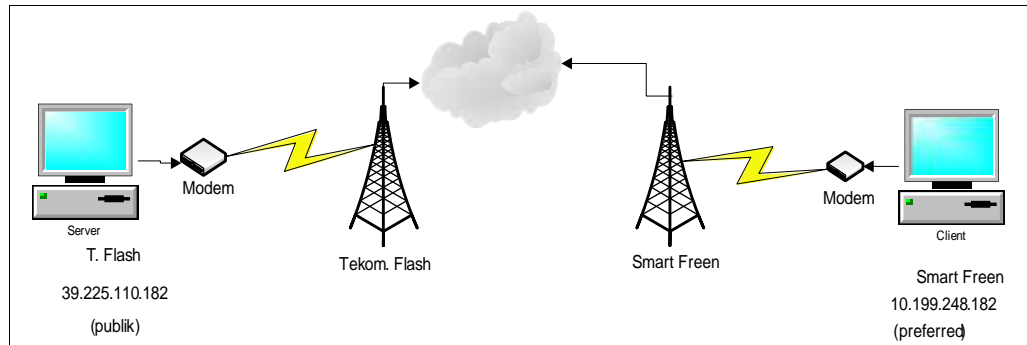


Gambar5.16 status koneksi TCP dari komputer Surya-PC sebagai klien, berhasil menghubungi server melalui LAN

Dari pengujian A dan B di atas, maka dapat disimpulkan untuk permainan Battleship melalui scenario 1, yaitu koneksi melalui LAN sebagai berikut:

- a. Koneksi berhasil dibangun dan permainan dapat berjalan dengan lancar melalui koneksi point-to-point, dengan syarat tidak ada firewall yang membatasi akses port 1234, dan setting IP address kedua komputer sudah benar (syarat ini harus terpenuhi, baik secara setting static maupun DHCP)
- b. Koneksi dari komputer klien ke server dapat menggunakan port TCP berapa saja, namun port yang dituju pada komputer server tetap terpantau pada port 1234. Hal ini karena port akses TCP dari komputer klien bersifat dinamik, yang diatur oleh sistem operasi pada komputer klien. Sedangkan pada komputer server, port yang digunakan diatur oleh program Battleship.

5.3.2 Skenario 2 pengujian melalui koneksi internet



Gambar 5.17 pengujian melauai jaringan Internet

Pada scenario pengujian ini, dilakukan koneksi melalui operator GSM 3G. Setting IP yang diperoleh secara DHCP dari operator GSM untuk Surya-PC Menggunakan IP address public : 39.225.110.182 Subnetmask 255.255.255.255 Nama operator : Telkomsel Flash Digunakan sebagai Server Game, karena mendapat IP Publik

```
PPP adapter HUAWEI3G.telkomflash:
Connection-specific DNS Suffix . : 
Description . . . . . : HUAWEI3G.telkomflash
Physical Address. . . . . : 
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . : Yes
IPv4 Address. . . . . : 39.225.110.182(Preferred)
Subnet Mask . . . . . : 255.255.255.255
Default Gateway . . . . . : 0.0.0.0
DNS Servers . . . . . : 192.168.98.28
                        10.11.12.14
Primary WINS Server . . . . . : 10.11.12.13
Secondary WINS Server . . . . . : 10.11.12.14
NetBIOS over Tcpip. . . . . : Disabled
```

Gambar 5.18 IP modem internet yang diperoleh secara otomatis melalui operator GSM pada komputer server (memperoleh ip public)

Setting IP yang diperoleh secara DHCP dari operator GSM untuk Yulyahr PC Menggunakan IP address private : 10.199.248.182 Subnetmask 255.255.255.255 Nama operator : Smart Fren Digunakan sebagai klien, karena mendapat IP private.

```
PPP adapter Wireless Terminal:
Connection-specific DNS Suffix . : 
Description . . . . . : Wireless Terminal
Physical Address. . . . . : 
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . : Yes
IPv4 Address. . . . . : 10.199.248.182(Preferred)
Subnet Mask . . . . . : 255.255.255.255
Default Gateway . . . . . : 0.0.0.0
DNS Servers . . . . . : 10.17.3.245
                        10.17.3.244
NetBIOS over Tcpip. . . . . : Disabled
```

Gambar 5.19 IP modem internet yang diperoleh secara otomatis melalui operator GSM pada komputer klien Yulyahr-PC (memperoleh ip private)

Setelah kedua komputer terhubung ke internet, maka dilakukan pengujian koneksi pada permainan Game Battleship. Gambar 5.20 dan 5.21 menunjukkan aplikasi permainan pada kedua komputer sedang menunggu koneksi.



Gambar 5.20 Server yang terkoneksi ke internet sedang menunggu koneksi dari komputer klien



Gambar 5.21 Klien Yulyahr-PC yang terkoneksi ke internet sedang menunggu koneksi dari komputer server

Setelah ditunggu beberapa lama, koneksi tidak juga terbentuk. Dilakukan monitoring terhadap status TCP yang dikirim oleh kedua komputer, seperti ditunjukkan oleh gambar 5.22 dan 5.23 berikut ini.

```
C:\Users\Surya>netstat
Active Connections

```

Proto	Local Address	Foreign Address	State
TCP	39.225.110.182:53096	212.96.161.239:http	TIME_WAIT
TCP	39.225.110.182:53098	106.10.199.10:http	ESTABLISHED
TCP	39.225.110.182:53102	118.98.38.6:http	TIME_WAIT
TCP	127.0.0.1:3979	Surya-PC:49158	ESTABLISHED
TCP	127.0.0.1:3979	Surya-PC:49159	ESTABLISHED
TCP	127.0.0.1:5357	Surya-PC:53100	TIME_WAIT
TCP	127.0.0.1:49158	Surya-PC:3979	ESTABLISHED
TCP	127.0.0.1:49159	Surya-PC:3979	ESTABLISHED
TCP	127.0.0.1:52126	Surya-PC:52127	ESTABLISHED
TCP	127.0.0.1:52127	Surya-PC:52126	ESTABLISHED
TCP	127.0.0.1:53100	Surya-PC:wsd	TIME_WAIT

```
C:\Users\Surya>ipconfig /all | more
```

Gambar 5.22 Status koneksi TCP pada komputer server yang terhubung ke internet

Gambar di atas menunjukkan bahwa sampai beberapa saat, koneksi dari komputer klien ke server melalui jaringan internet tidak berhasil juga. Hal ini diyakini karena adanya sistem firewall yang membatasi koneksi internet, yang tidak membolehkan koneksi dari dan menuju port 1234, untuk mengakses game multiplayer Battleship.

Bila ditinjau dari penelusuran rute koneksi (traceroute) dari komputer klien (Yulhar-PC) yang terkoneksi melalui IP private pada jaringan Smartfren, seperti yang ditunjukkan oleh gambar 5.23 di bawah ini, maka dapat diduga firewall dilakukan oleh salah satu server di internet, yaitu dengan IP address server 180.240.191.158 (Telkom Singapura), karena mulai dari server tersebut dan seterusnya, paket terhenti (RTO).

```
C:\Users\YULYA HR>netstat
Active Connections
Proto Local Address Foreign Address State
TCP 10.193.50.214:49205 124.155.223.96:http ESTABLISHED

C:\Users\YULYA HR>tracert 39.225.110.182
Tracing route to 39.225.110.182 over a maximum of 30 hops
  0  422 ms  219 ms  132 ms  10.17.94.85
  1  *      *      *      Request timed out.
  2  258 ms  125 ms  165 ms  10.17.97.41
  3  *      210 ms  385 ms  202.70.48.49
  4  98 ms  113 ms  131 ms  202.70.48.17
  5  *      561 ms  *      ip-179-13.moratelindo.co.id [202.43.179.13]
  6  *      *      *      Request timed out.
  7  166 ms  197 ms  253 ms  129.128.240.180.telin.sg [180.240.128.129]
  8  198 ms  158 ms  167 ms  14.190.240.180.telin.sg [180.240.190.14]
  9  *      397 ms  250 ms  86.190.240.180.telin.sg [180.240.190.86]
 10  148 ms  158 ms  313 ms  158.191.240.180.telin.sg [180.240.191.158]
 11  *      *      *      Request timed out.
 12  *      *      *      Request timed out.
 13  *      *      *      Request timed out.
 14  *      *      *      Request timed out.
 15  *      *      *      Request timed out.
 16  *      *      *      Request timed out.
 17  *      *      *      Request timed out.
 18  *      *      *      Request timed out.
 19  *      *      *      Request timed out.
 20  *      *      *      Request timed out.
```

Gambar 5.23 . Status penelusuran rute terhenti pada hop ke 8, yaitu salah satu router di Telkom Singapura

Dari pengujian scenario 2 ini, dapat disimpulkan bahwa:

- Pengujian koneksi melalui internet sangat tergantung dari apakah port 1234 dibuka di sepanjang jalur rute.
- Apabila dibuka, maka diduga kuat koneksi game Battleship melalui jaringan internet dapat dilakukan.

5.4 Pengujian *Black box*

Model atau cara pengujian pada *Game Battleship* ini yaitu menggunakan *Black Block*

5.4.1 Pengujin Menu Koneksi

Tabel 5.1 Pengujian menu utama game *Battleship*

Dekripsi	Prekondisi	Prosedure pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi	Hasil yang didapat	Kesimpulan
Pengujian Menu ouput	Tampilan Layar login aplikasi	1.Masukan <i>Ip</i> untuk m ulai 2.Klik tombol Mulai untuk memulai permainan	Data Username benar <hr/> Data User name kosong	Data berhasil di validasi tidak ad instruksi eror.	Layar yang ditampilkan sesuai dengan yang diharapkan	Data Berhasil divalidasi tidak ada intruksi error	Diterima

Berdasarkan dari hasil pengujian from menu koneksi diatas dengan menggunakan *black box* diatas dapat disimpulkan bahwa hasil yang diharapkan sesuai dengan yang diinginkan dan diterima

5.4.2 Pengujian input from nama

5.2 Tabel pengujian menu from nama

Dekripsi	Prekondisi	Prosedure pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi	Hasil yang didapat	Kesimpulan
Pengujian Menu from nama	Tampilan Layar login masukan nama	Klik tombol OK untuk memulai penyusunan kapal	Masukan kapal dalam kotak kotak yang disediakan	Kapal berhasil disusun tidak ada instruksi error.	Layar yang ditampilkan sesuai dengan yang diharapkan	Data Berhasil divalidasi tidak ada intruksi error	Diterima
Pengujian Menu from mulai	Tampilan Layar login main	Klik tombol mulai untuk memulai peperangan	Klik kanan mouse untuk mengubah kapal arah horizontal maupun vertikal	-	Layar yang ditampilkan sesuai dengan yang diharapkan	Data Berhasil divalidasi tidak ada intruksi error	Diterima

Berdasarkan dari hasil pengujian from nama dengan menggunakan *black box* diatas dapat disimpulkan bahwa hasil yang diharapkan sesuai dengan yang diinginkan dan diterima

5.4.3 pengujian tampilan medan perang

5.3 Tabel pengujian medan perang

Dekripsi	Prekondisi	Prosedure pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi	Hasil yang didapat	Kesimpulan
Pengujian Menu medan perang	Tampilan Layar untuk memulai perang	Klik kotak yang kita anggap letak kapal lawan.	Penembakan berhasil dan penembakan gagal Penembakan berhasil dan kapan terbakar	Tampilan kapal yang terbakar dan yang masuh utuh	Layar yang ditampilkan sesuai dengan yang diharapkan	Data Berhasil divalidasi tidak ada intruksi error	Diterima
Pengujian menu pesan	Tampilan menu pesan	Untuk pengiriman pesan	Tulis pesan untuk megirin kekotak lawan	Pesan yang ditulis berhsil diterima	Layar yang ditampilkan sesuai dengan yang diharapkan	Data Berhasil divalidasi tidak ada intruksi error	Diterima

Dari hasil pengujian *Black Box* didapatkan hasil bahwa mlalui pengujian berdasarkan *Black Box* ternyata mampu memberikan hasil yang optimal dan layak digunakan. Keluaaran yang diharapkan sesuai dengan yang diharapkan.

5.4.4 Kesimpulan Pengujian dengan *black box*

Untuk pengujian dengan *black box*, yaitu menguji sistem secara keseluruhan untuk menentukan kesalahan-kesalahan program dan antisipasi error program. Dari tabel pengujian maka dapat disimpulkan bahwa sistem yang dibuat telah bebas dari kesalahan.

5.5 Pengujian Sistem Kepada User (*User Acceptance Test*)

Pada pengujian *user acceptance test* ini dilakukan dengan melibatkan pengguna(*user*) secara langsung dimana pengguna ini merupakan orang awam bukan pakar yang ingin melakukan pengujian.

Setelah dilakukan pengujian sistem kepada pengguna, selanjutnya akan diberikan kuisisioner yang berisi pertanyaan-pertanyaan mengenai sistem yang telah digunakannya secara keseluruhan. Parameter yang digunakan pada kuisisioner ini adalah sebagai berikut:

1. Tampilan yang berminat memainkan Game
2. Tampilan sistem secara umum
3. Kemudahan akses pada menu-menu yang ada pada sistem
4. Penyajian informasi sistem
5. Manfaat sistem
6. Penilaian secara keseluruhan sistem

Kategori penilaian yang digunakan untuk kuesioner antara lain:

1. Amat Bagus
2. Bagus
3. Cukup
4. Kurang

Berikut hasil kuisisioner yang diberikan ke masing-masing responden yang dalam hal ini melibatkan 10 orang responden untuk menjawab pertanyaan yang telah diberikan.

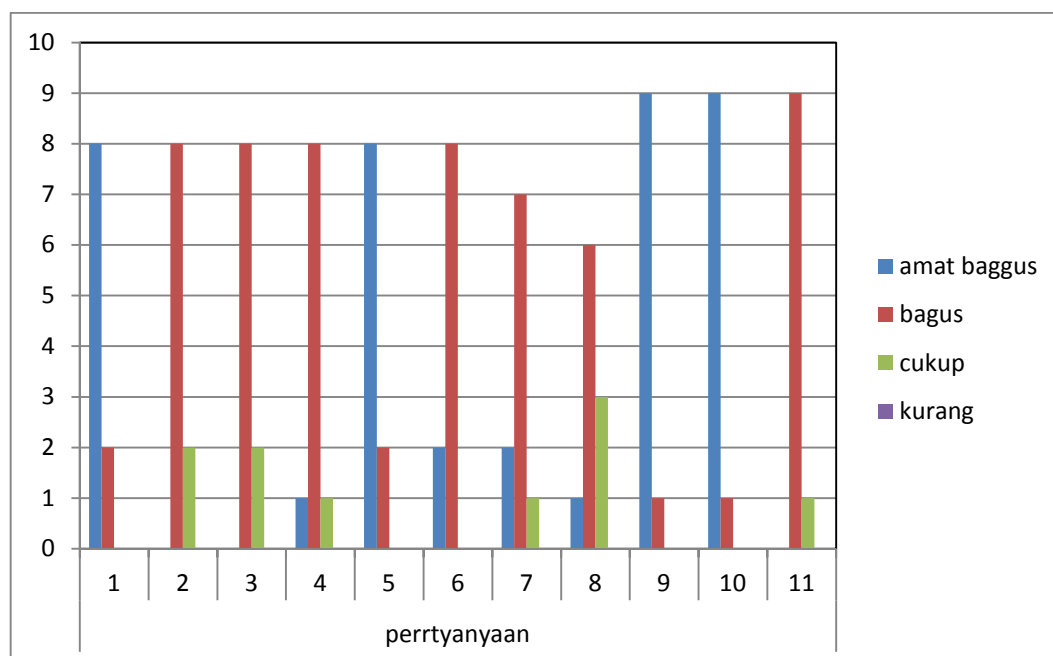
5.4 tabel pengujian kuisisioner

No	Pertanyaan	Jawaban			
		Amat bagus	Bagus	Cukup	Kurang
1	Setelah anda memainkan game ini, bagaimana menurut anda tentang permainan <i>battleship</i> ini	1	8	1	-
2	Bagaimana tanggapan anda dengan tampilan game <i>Battleship</i> ini	-	8	2	-
3	Bagaimana menurut anda dengan menu-menu pada game <i>Battleship</i> ini	-	8	2	-
4	Bagaimana menurut anda kelengkapan data pada game <i>battleship</i> ini	1	8	1	-
5	Bagaiman menurut anda game sebelumnya hanya bisa dimainkan dengan satu komputer, dan pada game ini biasa dimainkan lebih dari satu komputer.	8	2	-	-
6	Bagaimana pendapat anda kalau game ini digunakan oleh kalangan umum	2	8	-	-
7	Setelah anda mencoba permainan ini bagaimanakah dari tampilan hasil yang yang keluar	2	7	1	-
8	Setelah anda memainkan game Battleship ini bagaimanakah tanggapan anda terhadap tingkat kesulitan pada game ini	1	6	3	-
9	Bagaimana menurut anda bawa game battleship ini juga bisa untuk mengirim pesan antara komputer satu dengan komputer kedua	1	9	-	-
10	Bagaimana tanggapan anda dengan sound pada game battleship ini	9	1	-	-
11	Bagaimana Penilaian anda tentang <i>game</i> battleship ini secara keseluruhan	-	9	1	-

Tabel 5.5 Pengujian *User Acceptance Test*

Pertanyaan	Jawaban Mahasiswa							
	Amat bagus	%	Bagus	%	Cukup	%	kurang	%
1	8	80%	2	20%	0	0%	0	0%
2	0	0%	8	80%	2	20%	0	0%
3	0	0%	8	80%	2	20%	0	0%
4	1	10%	8	80%	1	10%	0	0%
5	8	80%	2	20%	0	0%	0	0%
6	2	20%	8	80%	0	0%	0	0%
7	2	20%	7	70%	1	10%	0	0%
8	1	10%	6	60%	3	30%	0	0%
9	1	10%	9	90%	0	0%	0	0%
10	9	90%	1	10%	0	0%	0	0%
11	0	0%	9	90%	1	10%	0	0%
Total	32	29%	68	61,8%	10	9%	0	0 %

Dari tabel 5.5 dapat digambarkan dalam bentuk grafik seperti dibawah ini:



Dari hasil penilaian pengujian *user acceptance test* dapat diambil kesimpulan yaitu:

1. Pengguna sistem yang telah memilih Amat Bagus mendapat nilai 29%
2. Pengguna sistem yang telah memilih Bagus mendapat nilai 61 %
3. Pengguna sistem yang telah memilih Cukup mendapat nilai 9 %
4. Pengguna sistem yang telah memilih Kurang mendapat nilai 0 %

5.5.1 Kesimpulan Pengujian dengan menggunakan *user acceptance test*

Untuk pengujian sistem kepada user secara langsung (*user acceptance test*), yaitu pengujian sistem secara langsung kepada pengguna yang selanjutnya akan diberikan kuisioner mengenai sistem yang digunakan. Kuisioner yang disebar yaitu terdiri dari 10 responden. Dengan hasil perhitungan persentase kuisioner tersebut, maka dapat disimpulkan bahwa sistem telah layak digunakan dan keluaran yang diberikan sistem sesuai dengan tujuan yang diharapkan.

BAB VI

PENUTUP

6.1 Kesimpulan

Setelah menyelesaikan permainan strategi *Battle Ship* yang dapat dimainkan pada jaringan, penulis menarik kesimpulan sebagai berikut :

1. Pembangunan *Game Battleship Multi Player* telah bisa dikembangkan dengan menerapkan tools OOP (*object oriented programming*) menggunakan bahasa pemrograman *visual basic* Namun, game ini memiliki kelemahan yaitu game ini tidak bisa dimainkan oleh satu komputer. Game ini hanya bisa dimainkan dengan menggunakan jaringan LAN.
2. Game ini dirancang dengan menggunakan OOA (*Object oriented analysis*) dan OOD (*object oriented design*).
3. Dari hasil pengujian, dapat disimpulkan bahwa game ini telah layak untuk diterapkan hal ini dapat dilihat dari hasil pengujian UAT dengan persentase nilai sebesar 80 % bagus. sehingga *game* ini dapat diterima oleh orang-orang yang suka main game.
4. *Game battleship* ini memungkinkan pemakai (*user*) komputer untuk memainkan permainan *Battle Ship* tanpa harus berhadapan secara langsung

6.2 Saran

Penulis memberikan beberapa saran sebagai berikut :

1. *Game battleship* ini dapat dikembangkan untuk *user* yang lebih banyak (lebih dari 2 orang).
2. *Background* pada *game battleship* ini masih sederhana, akan lebih baik menarik lagi apabila *background* terdiri dari animasi-animasi yang lebih bervariasi.
3. *Game Battleship* ini dapat ditambahkan konsep *Artificial Intelligence* (AI) sehingga dapat dimainkan dengan komputer.

DAFTAR PUSTAKA

- Asman, Dian Nuryahdi. 2012. *Analisis dan Perancangan Sistem Informasi Akademik Berbasis Client Server Dengan Metode Object Oriented Analysis And Design*. Jurusan Magistes Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Darma Palembang.
- Company, Milton Bradley. 1990. *Battleship*.
- Crawford, Chris. 1982. *The Art of Computer Game Design*.
- Djoko Pramono, Mudah menguasai *Visual Basic 6*, PT. Elex Media Komputindo, 2002.
- Fullerton, Tracy. 2008. *Game Design Workshop*. USA: Morgan Kaufmann Publisher.
- Husni. 2003. Implementasi Jaringan Komputer dengan Linux Redhat 9. Andi : Yogyakarta
- HM, Jogyianto. Pengenalan computer. Yogyakarta : penerbit Andi Yogyakarta, 2004
- <http://are-tech.blogspot.com/2009/02/sejarah-game.html>
- Muchtar, Dawang, 2005, Desain, Game, <http://gameworldamazing.com>, Wordpres. Com, Jakarta
- Munir, Rinaldi. 2009. *Diktat Kuliah IF2251 Strategi Algoritmik. Program Studi Teknik Informatika*. Sekolah Teknik Elektro dan Informatika : Institut Teknologi Bandung.
- Loguidice, Bill and Matt barton. 2009. *Vintage Games*. USA: Focal Press
- Julius, Hermawar, 2005, Analisa, Desaign & Programan, Berorientasi Objek dengan UML Dan Visua Basic, Andi, Yogyakarta
- Pramono, Djoko. 2002. *Mudah menguasai Visual Basic 6*. PT. Elex Media Komputindo : Jakarta.
- Stallings, W. *Local Network*, Macmillan Publishing Company, 1985.

Sumarsono, Abraham Ranardo. 2009. *Makalah IF3051 Strategi Algoritma. Penerapan Algoritma Runut Balik dalam Pencarian Teka Teki Battlehsip.* Sekolah Teknik Elektro dan Informatika : Institut Teknologi Bandung.

Thompson, Jim.2007. *The Computer Game Design Course.* London: Quarto Publishing.

Wijaya, Hartono Sulaiman. 2011. *Game Battleship Board Checking. School of Electrical Engineering and Informatics : Bandung Institute of Technology.*

Wibawanto, wandah,2005, membuat game dengan Visual Basic, Andi, Yogyakarta